Computing&AI Connect
ISSN: 3104-4719
2026, Article ID. x, Cite as: https://www.doi.org/10.69709/xxx

SCIFINITI PUBLISHING

OPEN ACCESS

Perspective

Fine-tuning vs RAG: A Position Paper from the Perspective of LLM-based Cybersecurity Modeling

Mohammad Amaz Uddin [1,2,*] **Iqbal H. Sarker** [3,*]

[1] Department of Computer Science and Engineering, International Islamic University Chittagong, Chattogram 4318, Bangladesh;

[2] Department of Computer Science and Engineering, BGC Trust University Bangladesh, Chittagong 4381, Bangladesh;

[3] Centre for Securing Digital Futures, Edith Cowan University, Perth 6027, WA, Australia;

Abstract

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) tasks, enabling critical capabilities in fields such as business and finance with their cybersecurity analysis. In particular, LLMs are becoming vital in cybersecurity in providing more accurate solutions to different types of threats, vulnerabilities, and security challenges through pattern recognition, intelligent response generation, and automating threat detection. Although LLMs are pre-trained with a large amount of knowledge, their application in any specific domain, such as cybersecurity, is often limited by the domain knowledge constraints of their pre-training data. To solve this problem, incorporating domain-specific knowledge is more often accomplished through fine-tuning or Retrieval-Augmented Generation (RAG). RAG enriches the model's responses at inference time by retrieving relevant external information, while fine-tuning embeds new knowledge directly into the model's parameters. In this position paper, we describe fine-tuning and RAG from the perspective of cybersecurity applications. Moreover, we also discuss the hybrid approach of fine-tuning and RAG and highlight proper best cases for when to select fine-tuning, RAG, or a hybrid approach based on the desired cybersecurity use case and operational constraints. Overall, this position paper aims to contribute to the ongoing discussion of LLM adaptation processes and best practices for the purpose of applying RAG and or fine-tuning.

Keywords:
large language models; natural language processing; retrieval-augmented generation; GraphRAG; cybersecurity

1. Introduction

Over the past few years, the rapid advancement of LLMs has significantly expanded Artificial Intelligence (AI) applications, especially in fields such as cybersecurity, where they automate and accelerate key tasks that traditionally required substantial human effort. Modern LLMs, notable examples include Google's Gemini and OpenAI's Generative Pre-trained Transformer (GPT) series, have transformed NLP by achieving high levels of contextual reasoning, text generation, and understanding [1]. During pre-training, LLMs are exposed to diverse datasets, enabling broad generalization and domain expertise. Their ability to understand, generate, and reason over complex information makes them valuable tools in cybersecurity, analyzing logs, emails, and network traffic to detect phishing attacks, malware, spam, or insider threats, often faster while reducing manual workload, thereby saving time and costs.

Language models are knowledgeable from diverse datasets [2], but struggle with uncommon, domain-specific, or inaccurate data [3] and can't incorporate new information or changing facts after training, as their knowledge is static. This leads to temporal degradation, outdated or inaccurate responses, and hallucinations. These issues are concerning in high-stakes fields like cybersecurity, where accuracy, adaptability, and up-to-date threat intelligence are crucial for tasks like threat detection, anomaly identi-

* **Corresponding Author:**
Iqbal H Sarker, Centre for Securing Digital Futures,
Edith Cowan University, Perth 6027, WA, Australia m.sarker@ecu.edu.au;
Tel.: +xx-xxx-xxx-xxxx

fication, and malware analysis. Researchers have focused on customizing LLMs for specific industries, such as cybersecurity, healthcare, finance, and others, while enhancing their access to current data to address these challenges. Two prominent methods have emerged: fine-tuning and RAG. RAG keeps the model up-to-date without altering its internal parameters by retrieving relevant external knowledge in real time, similar to In-Context Learning (ICL) [4], which boosts responses by adding information to the input rather than adjusting the model's weights. Conversely, fine-tuning involves retraining the model on specialized datasets to enable it to learn domain-specific knowledge.

RAG is more flexible and scalable, but fine-tuning increases accuracy within a certain area. Both approaches offer distinct advantages and trade-offs regarding computational cost, data requirements, update frequency, and deployment complexity. In this position paper, we aim to explore which methodologies are more suitable for enhancing LLMs in low-resource, high-stakes domains like cybersecurity. Specifically, we focus on:

- Detailed analysis of effective fine-tuning techniques for cybersecurity modeling.
- Highlighting effective RAG strategies for cybersecurity modeling.
- Describe the key challenges and risks within the study of fine-tuning and RAG.
- Analyzing the comparative impact of retrieval-based vs. fine-tuning strategies and the hybrid approach.

2. Understanding Fine-Tuning and RAG

Fine-tuning, RAG, and their evolving hybrid approach are key paradigms in the development and use of LLMs. In this section, we will explore the background of fine-tuning, RAG, and their hybrid approach, including their techniques and procedures.

2.1. Fine-Tuning

Fine-tuning is the process that involves continuous training on smaller, more specialized datasets and is needed to specialize a pre-trained LLM for domain-specific tasks [5]. It has multiple methodologies, techniques, and applications. We will discuss the methodologies, techniques, and applications of fine-tuning in the context of cybersecurity.

2.1.1. Learning Paradigms in Fine-Tuning

Supervised Fine-Tuning (SFT): SFT [6] adapts large language models to cybersecurity tasks by training them on labeled datasets that capture real security threats, such as malware samples [7], spam messages [8], network intrusion annotations [9], and others. SFT improves the model's ability to map inputs to correct outputs for tasks like threat classification and anomaly detection. Figure 1 illustrates this SFT process.

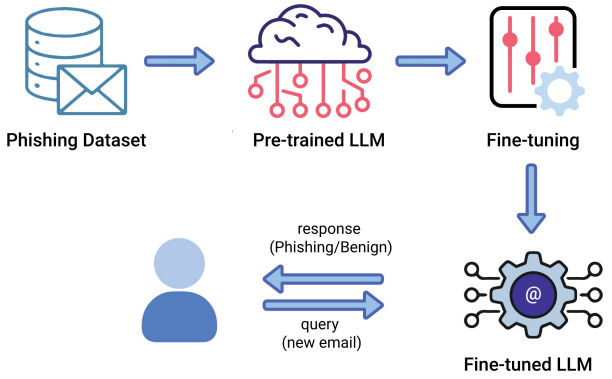

Figure 1: Supervised Fine-tuning in Phishing Email Detection.

Unsupervised Fine-Tuning (UFT): UFT [10] adapts pre-trained LLMs to the cybersecurity domain without labeled data by continuously training on large-scale unlabeled sources. This method utilises self-supervised learning and uses the underlying structure of the data to allow the model to learn cybersecurity terminology, patterns, and context.

Reinforcement Learning (RL): RL, particularly Reinforcement Learning from Human Feedback (RLHF) [11], offers a new frontier for the ongoing improvement of LLMs in cybersecurity through iterative feedback aligned with expert analyst preferences and operational goals. Different RL algorithms, such as reward modeling, Proximal Policy Optimization (PPO), and Direct Preference Optimization (DPO), enable adaptive learning in dynamic threat environments, supporting more effective security decision-making.

2.1.2. Fine-Tuning Techniques for Cybersecurity Tasks

Hyperparameter Tuning: Hyperparameter tuning is an effective technique for fine-tuning LLMs for improving model performance, generalization, and training stability. Selecting optimal hyperparameters, such as batch size, epochs, learning rate, and regularization, can significantly improve the model and prevent overfitting, especially in cybersecurity tasks where data are often noisy or imbalanced.

Parameter-Efficient Fine-Tuning (PEFT): PEFT [12] adapts LLMs to various cybersecurity tasks by updating a small number of model parameters, significantly reduc-

2024, Vol. 1, Issue 1
doi:10.x/journal.x.x.x

ing computation and memory resources while preserving pre-trained knowledge. Techniques like Low-Rank Adaptation (LoRA), adapters, and prompt tuning allow rapid model adaptation, which is useful for keeping pace with continuously changing cyber threats.

Task-Specific Fine-Tuning: Task-specific fine-tuning is the process of adapting a pre-trained language model to a specific cybersecurity task using labeled, domain-specific data to enable the model to identify task-relevant patterns and enhance its performance. Although it leads to catastrophic forgetting due to parameter updates, task-specific fine-tuning can help the model to achieve high accuracy in targeted cybersecurity applications.

Transfer Learning: Transfer learning adapts pre-trained LLMs to cybersecurity tasks using limited data, with cybersecurity domain-specific models, such as SecureBERT [13], achieving superior performance using fewer computational resources compared to out-of-the-box LLMs in malware, phishing, and vulnerability detection.

Few-Shot and Zero-Shot Fine-Tuning: Few-shot and zero-shot fine-tuning approaches leverage the generalization capabilities of large pre-trained models to conduct tasks with few or no labeled input [14]. Few-shot learning enables the model to infer patterns in new data for detection using a small number of labeled examples. In contrast, zero-shot learning relies solely on prompts and prior knowledge, without the use of task-specific examples.

2.2. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) [15] is an emerging paradigm that augments LLMs with knowledge retrieval and generation capabilities. While LLMs demonstrate remarkable performance, they face issues such as hallucinations and outdated information, especially with Out-of-Distribution (OoD) questions. The hallucination issue arises when a model produces information that seems correct and looks real, but is factually incorrect or fabricated. RAG overcomes these challenges by dynamically retrieving up-to-date, domain-specific information from structured and unstructured sources at inference time [16]. Unlike traditional fine-tuning, which encodes task-specific knowledge into model parameters, RAG, particularly in the cybersecurity domain, accesses external sources such as threat databases, security logs, and technical documentation, ensuring adaptability, interpretability, and new information without retaining the model.

2.2.1. RAG Core Components & Workflow

Core Components: RAG is composed of multiple components that work together to generate context-aware responses. The first component is the retriever, which identifies and collects relevant documents and knowledge snippets from external sources such as databases, search indexes, or structured knowledge bases, with key resources in cybersecurity including such as MITRE ATT&CK [17], Common Vulnerabilities and Exposures (CVE) [18], and National Vulnerability Database (NVD) [19]. To support semantic retrieval, embedding models convert both queries and documents into vector representations to facilitate dense retrieval of the most contextually relevant information from a knowledge base. The knowledge base is another component that serves as an external repository of domain-specific information, including texts, databases, APIs, and real-time feeds stored in vector databases. Finally, the generator, typically a language model, produces the final response by combining the original query and the retrieved context to create accurate answers.

RAG Workflow: The workflow of RAG is based on the input query, retrieval, and generation process. The working procedure begins with a user input query, which is converted into a vector to represent the semantic meaning. This vector is used to search for the most appropriate document chunks within a vector database for the user-submitted input query. After that, the retrieved document chunks are filtered and ranked to select the best matches to the input query. Finally, a language model processes a prompt formed by combining the retrieved context with the original query, generating a coherent and contextually appropriate response. The user then receives the final formatted output.

The RAG workflow can be understood through an example of misleading feedback or fake review detection in Small and Medium Enterprises (SMEs) [20]. These fake or deceptive reviews are a growing concern in the SME industry [21], and RAG deployment can play a significant role in this industry. Using the RAG methodology, SME analysts can identify and detect fake reviews more accurately. Firstly, submit a query such as a product review with the question “Is this review fake?”, and the system converts the review text into a semantic vector that captures its meaning. Then the semantic vector is used to search for the labeled fake reviews, expert analysis, and the most relevant flagged document chunks in the database. Those relevant document chunks, such as reviews with a similar pattern, are retrieved. The retrieved context is then combined with the original query and passed to a generation model, which generates an appropriate answer. An overview of the RAG workflow is shown in Figure 2 for fake review detection.

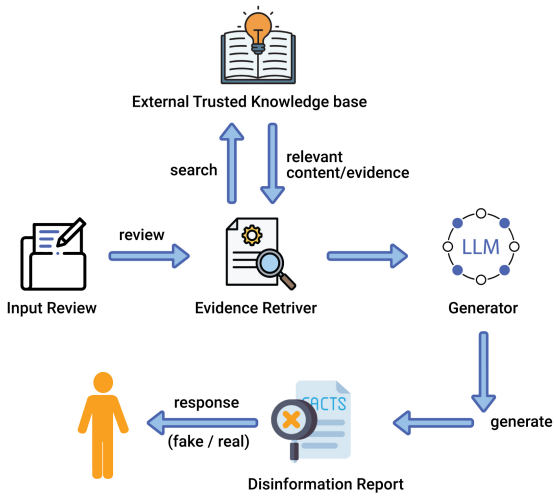

Figure 2: RAG Workflow in Disinformation Detection.

2.2.2. Graph Retrieval-Augmented Generation (GraphRAG)

GraphRAG is a graph-based RAG variant or successor of RAG that integrates Knowledge Graphs (KGs) into the RAG pipeline to improve reasoning, contextual understanding, and explainability in LLMs [22]. The GraphRAG framework is designed to overcome the drawbacks of the traditional RAG system, including a lack of understanding of relationships, failure in multi-hop reasoning, and a tendency to hallucinate. The workflow of GraphRAG consists of several key steps. For example, fake review detection in SMEs, when an SME analyst submits a query such as a product review with a similar question, "Is this product review fake?", the system first converts the review text into structured entities (user, product, review, sentiment) and relationships. These entities and relationships are stored in a Knowledge Graph that can be constructed from both local data sources, such as private reviews, customer histories, and external data sources (public review repositories or others). When the query is processed, the system performs a graph-based retrieval, rather than relying solely on text similarity. It traverses the knowledge graph to identify relational patterns, such as users posting multiple identical reviews, repeated sentiment expressions across unrelated products, or unnatural temporal correlations. The retrieved subgraph representing the most relevant entities and their relationships is aggregated with the original query and submitted to the language model. Finally, the model generates a response by reasoning over the retrieved graph context, enabling context-aware and explainable detection of fake reviews. The example workflow of GraphRAG is shown in Figure 3.

In comparison with other domains, this framework is not explored too much in the cybersecurity domain, but day by day, it is becoming popular in this domain. For example, in [23], the authors proposed the CyKG-RAG framework, which integrates KGs with the RAG approach for cybersecurity threat detection and analysis. Similarly, in other research areas such as network security monitoring, analysis, GraphRAG creates its demands.

2.3. Hybrid Approach

The hybrid approach, which combines fine-tuning and RAG, has gained popularity since it has addressed the drawbacks of each technique when used independently. Figure 4 provides a conceptual overview of the hybrid approach combining fine-tuning and RAG.

This approach is beneficial when there is a necessity for domain-specific knowledge in combination with a real-time updating process. For instance, in cybersecurity modeling, both deep domain adaptation and real-time responsiveness might be needed, and this hybrid methodology would enable an intelligent and powerful approach to modeling behavior. The hybrid approach has also been successfully applied in various other fields, including healthcare [24], multilingual question answering [25], etc.

The hybrid approach provides several key benefits, including enhanced robustness, lower retraining costs, explainability, and task flexibility. For instance, spam filtering in email, messaging, and other communication channels requires knowledge of both linguistic patterns and dynamic content structures. The fine-tuned model is adjusted to identify linguistic and structural traits of spam, such as unusual sender behavior and excessive use of specific phrases. By acquiring the latest known spam samples, blacklisted IPs (Internet Protocols), or domains, or publicly available spam databases, the RAG module enhances the detection process and enables real-time, explainable judgments based on both live data and learned patterns. This hybrid approach increases detection accuracy and facilitates better generalization to multilingual or obfuscated spam tactics and zero-day spam patterns.

3. Applications in Cybersecurity

In this section, we outline several real-world applications in the cybersecurity domain where potential uses of fine-tuning and RAG methods for LLMs can make substantial contributions.

Phishing Detection and Analysis: Fine-tuned LLMs effectively detect phishing content by adapting to new phishing patterns through parameter-efficient and few-shot fine-tuning, while hyperparameter optimization fur-

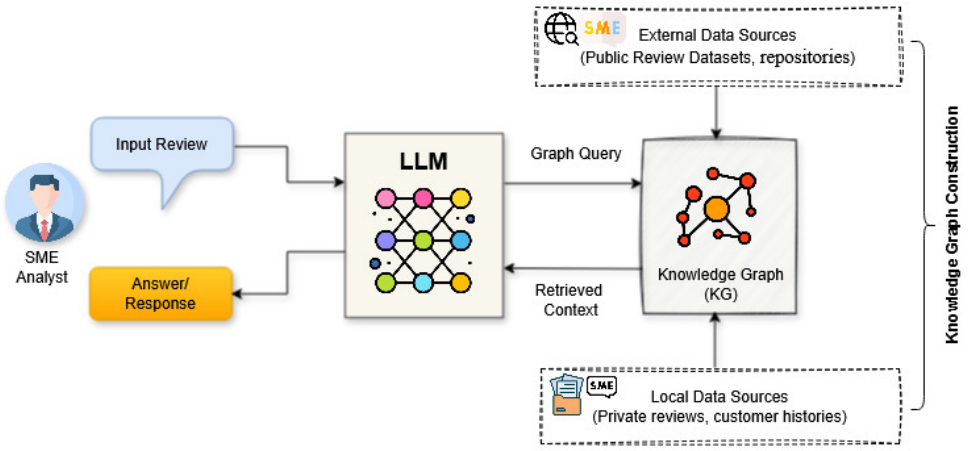

Figure 3: GraphRAG Workflow for Fake Product Review Detection in the context of SMEs.

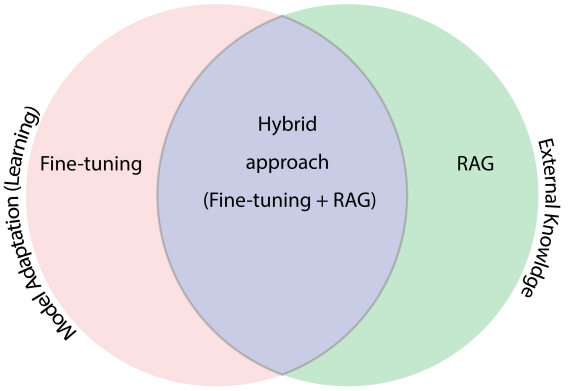

Figure 4: Hybrid Approach.

ther enhances detection accuracy. On the other hand, RAG works in scenarios where models need to take into account the most recent phishing samples or attack indicators from external sources, such as threat feeds or phishing databases, to adapt to emerging threats.

Intrusion Detection and Log Analysis: Supervised fine-tuning on datasets like UNSW-NB15, NSL-KDD, and CIC-IDS enables LLMs to detect intrusion activities in system logs or network traffic. Their contextual understanding aids in analyzing data, detecting rare attack vectors, and linking multi-step intrusions. RAG methods assess recent threats by retrieving up-to-date intelligence to enhance detection speed, accuracy, and organizational insights.

Vulnerability Detection and Analysis: Fine-tuned LLMs improve vulnerability management by summarizing complex CVE data and identifying key threats like buffer overflows and Remote Code Execution (RCE), aiding prioritization. RAG techniques, like the knowledge-level framework, retrieve relevant info from a vulnerability knowledge base to detect issues and boost accuracy.

Fraud Detection: Fine-tuned LLMs are ideal for identifying known and stable fraud patterns by incorporating temporal and behavioural features from a large volume of transactional data. RAG-based frameworks are useful for detecting fraud or deception across transactions because they can analyze transactions in real time and retrieve relevant trends, rules, and cases from associated fraud history. This performs exceptionally well in few-shot learning situations within domain-agnostic settings, so they counter new fraud schemes effectively.

SME Cybersecurity Advisor: SMEs are vulnerable to cyber threats due to limited resources, outdated systems, and a lack of security experts, leading to financial loss, loss of customer confidence, and operational interruption in those SMEs. Fine-tuned LLMs can act as cybersecurity advisers by learning from policies, processes, and event data from the domain, while RAG enhances this capability by retrieving real-time threat intelligence, compliance requirement updates and mitigating best practices, enabling cost-effective cybersecurity support without retraining.

4. Challenges and Risks

While both fine-tuning and RAG are effective approaches for adapting LLMs to cybersecurity tasks, they present distinct challenges and operational risks. Understanding the limitations of each method is essential for selecting or designing effective LLM-based cybersecurity solutions. Below, we summarize the key challenges and risks:

Data Dependency: Fine-tuning needs an extensive quantity of high-quality labeled data, which can often be challenging to obtain in specialized domains like cybersecurity due to security concerns or other constraints. RAG depends heavily on the external retrieval data qual-

ity and reliability, and raises the possibility of including poisonous, out-of-date, or biased data in its responses.

Security and Adversarial Risks: Fine-tuning faces security risks, such as data poisoning and hidden triggers. Attackers can exploit vulnerabilities in the model's decision process. RAG is at risk from attacks on its retrieval system, like injecting malicious data or tampering with knowledge bases, which can lead to harmful outputs.

Performance and Forgetting: Fine-tuning is fast but can overfit or forget earlier tasks if data is limited. RAG is slower due to external database retrieval and relies on database quality, risking performance drops if databases are unavailable or slow.

Maintenance and Resource Intensiveness: Fine-tuning requires periodic retraining, which is resource-intensive, especially for large models. Organizations must weigh the costs of computing and data annotation against accuracy improvements. RAG systems demand ongoing maintenance of retrieval databases and strong access controls to prevent data corruption and ensure integrity. Scaling the retrieval infrastructure for high query volumes can also be challenging.

Deployment Challenges: Fine-tuning large models for commercial use demands high compute power and strict data governance. Key challenges include model drift, repeatability, privacy compliance, and deployment complexities like explainability, version control, and domain bias. Deploying RAG commercially raises data security and governance risks when using internal sources. Low latency, scalability, and retrieval accuracy are difficult with large, changing datasets. The hybrid workflow also challenges data freshness, compliance, and explainability.

5. Our Position and discussion

Based on the comparative analysis of fine-tuning and RAG, in the context of cybersecurity modeling, adapting LLMs requires working with both approaches. Although each method offers distinct advantages, choosing between fine-tuning and RAG mainly depends on the selected topic use case, available resources, and operational requirements. Moreover, RAG is the preferred option for most enterprise use cases, while fine-tuning serves as the most practical solution for some scenarios. However, this does not mean that RAG is always the right choice over fine-tuning. To determine which method is best in the context of LLMs for cybersecurity modeling, it is essential to understand the different aspects and features.

Knowledge Integration: Fine-tuning absorbs and assimilates domain knowledge during training, resulting in high performance in this domain. However, it is static by nature and quickly becomes outdated without retraining, making it unsuitable for rapidly evolving fields like cybersecurity. In contrast, RAG ensures that the generated responses are constantly up to date by dynamically accessing external knowledge, but its reliance on retrieval can lead to discrepancies and errors in outputs.

Adaptability and Flexibility: Fine-tuning modifies the model's parameters through retraining using domain-specific data, which is an additional cost for any organization. RAG provides superior flexibility by retrieving relevant data from an external knowledge base without changing the model's internal parameters, though this increases complexity due to dependence on retrieval quality.

Latency: RAG is slightly slower due to the heavy data retrieval and generation phases, ensuring up-to-date data, which requires additional time. Fine-tuning is typically faster than RAG because it can generate answers almost instantly without needing to access external data sources.

Scalability: RAG is highly scalable, seamlessly handling large volumes of data by updating the external knowledge base without modifying the model itself. Fine-tuning is less flexible and encounters challenges when scaling for evolving domains; it needs retraining. Adapting to new information requires retraining or additional fine-tuning, thereby limiting responsiveness in fast-evolving domains.

Computational Resources: RAG has cheaper training costs because it leverages the base or pre-trained language model and doesn't require any modifications during the training phase. However, computer resources are needed for retrieval during inference. On the other hand, due to the resource requirements and dataset preparation, fine-tuning during the training phase necessitates a large amount of memory, time, and numerous high-performance GPUs.

Additionally, the hybrid option can be a suitable solution on some occasions. Hybrid methods use the fixed domain adaptation value gained through fine-tuning, while also utilizing the dynamic adaptation provided by RAG, creating systems that are knowledgeable and contextual. But without careful planning, the hybrid system may potentially inherit the drawbacks of both strategies, including potential latency bottlenecks, difficult implementation, and expensive maintenance. A comparison of fine-tuning, RAG, and the Hybrid Approach is presented in Table 1, focusing on various aspects to choose the most suitable one for a particular use case.

Moreover, nowadays, another technique named prompt engineering is also used to design and optimize prompts to effectively guide LLMs, particularly in NLP tasks, in generating desired outcomes or responses [26]. It can be another solution to pick up instead of selecting fine-tuning or RAG in cybersecurity-based experiments. It is a

Table 1: Comparison of Fine-Tuning, RAG, and Hybrid Approach for quick decision-making

Aspect	Fine-tuning (FT)	RAG	Hybrid
Core Idea	Update the model's parameters using domain/task-specific data.	Augments generation with retrieved relevant external documents.	FT + RAG: Domain-adapted model + real-time external retrieval.
Knowledge Source	Static	Dynamic	Both
Latency	Low	Medium	high
Transparency	Low	High	Medium
Scalability	Poor	Highly scalable	Good
Handles New Data	Requires retraining.	Instant: can update knowledge sources	Instant (update DB) + FT for deeper tuning
Cost	High	Moderate	High

faster, cost-effective, and lightweight approach compared to fine-tuning and RAG, and it works without retaining the model with minimal computational resources. In the cybersecurity domain, using prompt engineering can analyze malicious logs, detect phishing & fraud, disinformation, vulnerability assessment, etc. Although it has so many advantages, but also needs to keep in mind some limitations of this prompt engineering, such as no real-time detection, pre-trained knowledge, so the model can only respond based on previous knowledge.

6. Conclusion

This position paper highlights the role of LLMs in cybersecurity modeling by exploring two prominent approaches: fine-tuning and RAG. This study began by detailing various fine-tuning paradigms and techniques for cybersecurity modeling, highlighting their effectiveness and strengths in adapting LLMs for domain-specific tasks such as detection, classification, and analysis. The study also explored RAG with its core components, working procedure, and a variant. Along with these approaches, a hybrid approach that integrates fine-tuned LLMs with RAG mechanisms is also discussed. The hybrid integration of both paradigms introduces a promising method that combines dynamic reasoning over recently retrieved information with stable core knowledge. Moreover, this study presented the applications and challenges with the risk factors of fine-tuning and RAG. Ultimately, the choice between fine-tuning, RAG, or a hybrid should be guided by specific cybersecurity use cases, available data, infrastructure constraints, and desired levels of interpretability. We conclude that future cybersecurity systems will benefit most from models that combine the precision of fine-tuning with the adaptability of RAG, providing both strong baseline intelligence and real-time responsiveness to emerging threats.

Author Contributions

Conceptualization: M.A.U., I.H.S.; writing—original draft preparation: M.A.U. software and visualization: M.A.U.; Supervision: I.H.S. All authors reviewed and accepted the final version of the manuscript.

Funding

No external funding was received for this research.

Acknowledgments

AI tool (Grammarly) is used to refine the text, improving the languageprecisionandreadabilityofthe paper.

Data and materials availability

Not applicable.

Conflicts of interest

The author(s) declare no conflicts of interest regarding this manuscript.

References

[1] Iqbal H Sarker. Llm potentiality and awareness: a position paper from the perspective of trustworthy and responsible ai modeling. *Discover Artificial Intelligence*, 4(1):40, 2024.

[2] Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on*

Knowledge and Data Engineering, 36(4):1413–1430, 2023.

[3] Muhammad Arslan, Saba Munawar, and Christophe Cruz. Business insights using rag–llms: a review and case study. *Journal of Decision Systems*, pages 1–30, 2024.

[4] Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag. In *First Conference on Language Modeling*, 2024.

[5] Samar Pratap, Alston Richard Aranha, Divyanshu Kumar, Gautam Malhotra, Anantharaman Palacode Narayana Iyer, and Shylaja SS. The fine art of fine-tuning: A structured review of advanced llm fine-tuning techniques. *Natural Language Processing Journal*, page 100144, 2025.

[6] Aldo Pareja, Nikhil Shivakumar Nayak, Hao Wang, Krishnateja Killamsetty, Shivchander Sudalairaj, Wenlong Zhao, Seungwook Han, Abhishek Bhandwaldar, Guangxuan Xu, Kai Xu, et al. Unveiling the secret recipe: A guide for supervised fine-tuning small llms. *arXiv preprint arXiv:2412.13337*, 2024.

[7] Tristan Carrier. Detecting obfuscated malware using memory feature engineering. 2021.

[8] Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262, 2011.

[9] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE, 2015.

[10] Korawat Tanwisuth, Shujian Zhang, Huangjie Zheng, Pengcheng He, and Mingyuan Zhou. Pouf: Prompt-oriented unsupervised fine-tuning for large pre-trained models. In *International conference on machine learning*, pages 33816–33832. PMLR, 2023.

[11] Nathan Lambert. Reinforcement learning from human feedback. *arXiv preprint arXiv:2504.12501*, 2025.

[12] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mohammad Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[13] Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. Securebert: A domain-specific language model for cybersecurity. In *International Conference on Security and Privacy in Communication Systems*, pages 39–56. Springer, 2022.

[14] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 3816–3830, 2021.

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[16] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[17] The MITRE Corporation. Mitre att&ck®, 2025. Accessed: 2025-10-28.

[18] The MITRE Corporation. Common vulnerabilities and exposures (cve), 2025. Accessed: 2025-10-28.

[19] National Institute of Standards and Technology (NIST). National vulnerability database (nvd), 2025. Accessed: 2025-10-28.

[20] Iqbal H Sarker, Helge Janicke, Ahmad Mohsin, and Leandros Maglaras. Sme-team: Leveraging trust and ethics for secure and responsible use of ai and llms in smes. *arXiv preprint arXiv:2509.10594*, 2025.

[21] Ronnie Das, Wasim Ahmed, Kshitij Sharma, Mariann Hardey, Yogesh K Dwivedi, Ziqi Zhang, Chrysostomos Apostolidis, and Raffaele Filieri. Towards the development of an explainable e-commerce fake review index: An attribute analytics approach. *European Journal of Operational Research*, 317(2):382–400, 2024.

[22] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*, 2024.

[23] Kabul Kurniawan, Elmar Kiesling, and Andreas Ekelhart. Cykg-rag: Towards knowledge-graph enhanced retrieval augmented generation for cybersecurity. 2024.

[24] Bhagyajit Pingua, Adyakanta Sahoo, Meenakshi Kandpal, Deepak Murmu, Jyotirmayee Rautaray, Rabindra Kumar Barik, and Manob Jyoti Saikia. Medical llms: Fine-tuning vs. retrieval-augmented generation. *Bioengineering*, 12(7):687, 2025.

[25] Leandro Yamachita da Costa, João Baptista de Oliveira, et al. Adapting llms to new domains: A comparative study of fine-tuning and rag strategies for portuguese qa tasks. In *Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL)*, pages 267–277. SBC, 2024.

[26] Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, pages 387–402. Springer, 2023.

2026, Vol. 3
doi:10.x/journal.x.x.x

SCIFINITI
PUBLISHING

List of abbreviations

Large Language Models (LLMs)
Natural Language Processing (NLP)
Retrieval-Augmented Generation (RAG)
Generative Pre-trained Transformer (GPT)
In-Context Learning (ICL)
Supervised Fine-Tuning (SFT)
Parameter-Efficient Fine-Tuning (PEFT)
Unsupervised Fine-Tuning (UFT):
Reinforcement Learning (RL)
Reinforcement Learning from Human Feedback (RLHF)
Proximal Policy Optimization (PPO)
Direct Preference Optimization (DPO)
Low-Rank Adaptation (LoRA)
Common Vulnerabilities and Exposures (CVE)
National Vulnerability Database (NVD)