ISSN: 2664-9640

ARC-Guard: A High-Precision Multi-Agent Collaborative Framework for Email Classification based on RAG and CoT

Jiajun Deng[a], Yuanqing Xian[*]

School of Mathematics and Computer Science, Guangdong Ocean University, Zhanjiang 524088, China

[a]jiajundeng88@hotmail.com, [*]xianyuanqing@163.com

Abstract

As a critical infrastructure for personal and professional communication, email is facing increasingly severe threats from sophisticated phishing and spam campaigns, making robust email classification systems essential. Traditional classification methods struggle with semantic nuances, while standard Large Language Models (LLMs) often suffer from hallucinations or lack domain-specific context. To address these challenges, we propose ARC-Guard, a multi-agent framework specifically designed for high-precision email classification, which integrates Retrieval-Augmented Generation (RAG) and Chain-of-Thought (CoT) reasoning. The system comprises three dedicated agents: an Initial Analysis Agent for surface-level inspection, a Dual-Path RAG Agent for vector retrieval of similar historical emails, and a Chain-of-Thought Agent that synthesizes retrieved contexts to generate interpretable verdicts. Evaluations on the SecEmail dataset show that ARC-Guard achieves a state-of-the-art (SOTA) accuracy of 90.42%, significantly outperforming baseline models. These results demonstrate that combining retrieval mechanisms with step-by-step reasoning substantially enhances the robustness and interpretability of email threat detection.

Keywords

Email Classification; Large Language Models; Multi-Agent; Retrieval-Augmented Generation; Chain-of-Thought.

1. Introduction

With the rapid development of digital communication, email has become a primary vector for cyberattacks, particularly phishing and spam campaigns that exploit social engineering vulnerabilities. Therefore, developing robust, interpretable, and high-precision email classification systems has become a cornerstone of modern cybersecurity defense. Traditional detection methods predominantly rely on feature engineering and supervised learning, which often struggle to maintain generalization capabilities in dynamically adversarial threat environments. In recent years, the emergence of Large Language Models (LLMs) has brought a paradigm shift to this field. Leveraging exceptional semantic understanding and In-Context Learning (ICL) capabilities, LLM-based agents are gradually being integrated into automated Security Operations Centers (SOCs) to augment or replace rule-based systems [1],[2].

However, directly deploying general-purpose LLMs in high-stakes security decision-making scenarios still faces severe challenges. As highlighted in the pioneering work on Reflexion, language agents are highly prone to hallucinations and reasoning fallacies in the absence of an external feedback loop or reliable contextual anchors [3]. In the context of email security, this limitation manifests as a lack of knowledge regarding emerging threat patterns (knowledge cutoff) or an inability to infer malicious intent from subtle textual clues. Relying solely on the

ISSN: 2664-9640

parametric memory of LLMs often leads to overconfident yet erroneous judgments, failing to meet the stringent requirements for accuracy and transparency in practical deployments.

To address these challenges, we propose a novel multi-agent collaborative framework named ARC-Guard (Analysis, Retrieval, and CoT for Email Guarding). Inspired by the cognitive workflow of security experts, ARC-Guard coordinates three dedicated agents to achieve a synergistic leap in detection performance. The Initial Analysis Agent first conducts a panoramic scan of the email content to extract salient malicious indicators, establishing a foundational understanding of the threat landscape. Subsequently, the Retrieval-Augmented Generation (RAG) Agent serves as a knowledge bridge, employing a dual-path retrieval strategy parallelizing sparse (TF-IDF) and dense (Embedding) methods to precisely recall historical precedents, effectively filling knowledge blind spots [4],[5]. Finally, the Chain-of-Thought (CoT) Agent acts as the decision-making core, utilizing the deep reasoning capabilities of cutting-edge models such as Qwen-Plus and DeepSeek-V3 to execute explicit logical deduction based on multidimensional evidence.

To rigorously evaluate the effectiveness of ARC-Guard, we construct and release a high-quality benchmark dataset, SecEmail. We selected 1,200 high-quality unique samples from over 300,000 emails acquired online. This data has undergone rigorous cleaning, deduplication, and re-annotation processes, ensuring a balanced distribution of Ham, Phishing, and Spam categories. SecEmail is specifically designed to challenge the performance of LLMs under complex social engineering attacks, providing a robust testbed for our experiments. Based on this benchmark, we design a comprehensive experimental matrix comprising 12 configurations, comprehensively comparing the performance of Qwen-Plus and DeepSeek under No-RAG, Sparse RAG, and Dense RAG strategies, as well as in Direct Reasoning and Deep Thinking modes. This rigorous design aims to dissect the marginal contribution of each component to the final classification performance.

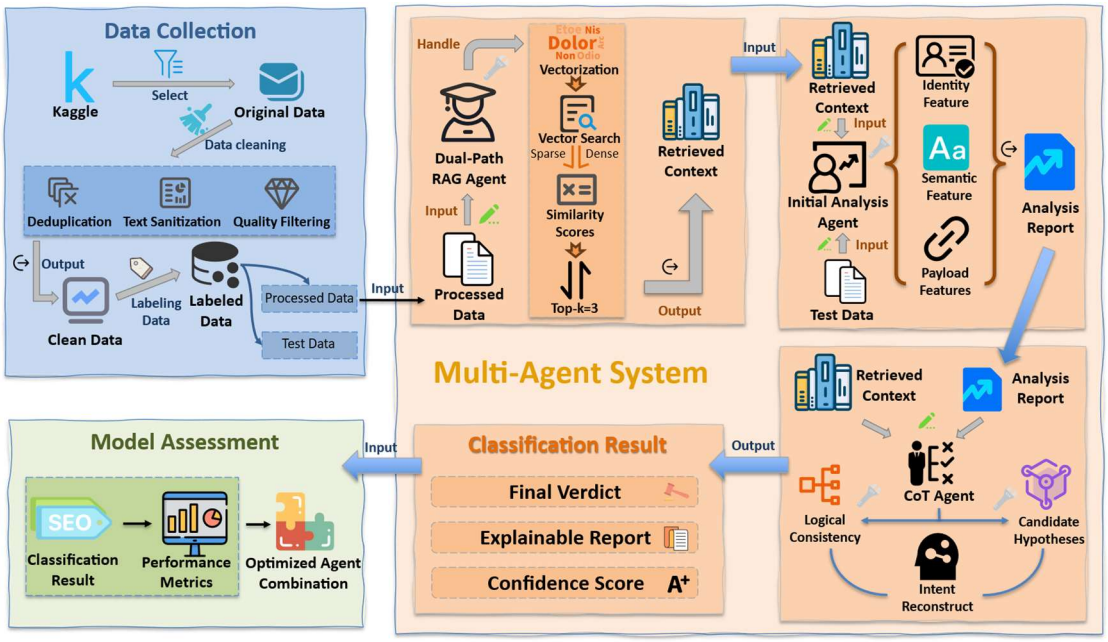

Figure 1. Overview of our multi-agent framework

Our extensive empirical analysis reveals compelling findings. The introduction of the RAG mechanism plays a foundational role in performance enhancement, increasing the baseline accuracy of both models from approximately 50% to over 70%, thereby validating the critical role of external knowledge in mitigating hallucinations. Furthermore, the integration of the

ISSN: 2664-9640

"Deep Thinking" mode triggers a substantial performance leap, yielding an additional accuracy improvement of 15-18%. Notably, the Qwen-Deep-SparseRAG configuration achieves a SOTA accuracy of 90.42%, demonstrating its superior robustness compared to dense retrieval methods when processing the discrete textual features typical of email data.

Our contributions are summarized as follows:

(1) We propose ARC-Guard, a novel multi-agent collaborative framework that effectively mitigates the hallucination and shallow reasoning issues of LLMs in security tasks by synergizing initial analysis, RAG, and CoT reasoning.

(2) We construct SecEmail, a rigorously filtered benchmark dataset derived from a large-scale public corpus, which fills the gap in existing datasets by providing high-quality, fine-grained samples of complex social engineering attacks.

(3) We conduct extensive empirical analyses that quantitatively validate the synergistic effects of sparse retrieval and deep reasoning, providing reusable best practices for deploying LLMs in vertical security domains.

2. Related Work

2.1. Email Topic Classification

Email classification encompasses tasks such as spam filtering, phishing detection, and topic categorization, representing a fundamental problem in cybersecurity and information retrieval. Traditional methods primarily rely on rule-based systems (e.g., SpamAssassin) and feature engineering combined with machine learning classifiers (e.g., Support Vector Machines (SVM) and Random Forests). Although computationally efficient, these methods often fall short when faced with dynamically changing email content, particularly against adversarial obfuscation and continuously evolving social engineering tactics [6],[7].

With the advent of deep learning, models such as CNNs, LSTMs, and BERT have significantly improved classification accuracy by capturing sequential dependencies and semantic contexts [8],[9]. However, these discriminative models typically operate as "black boxes," lacking the interpretability required for high-stakes security decisions [10]. Recently, Large Language Models (LLMs) have emerged as powerful tools for this task. Studies indicate that LLMs can leverage their vast pre-trained knowledge to perform zero-shot or few-shot classification of complex email intents [11],[12],[13]. Despite the promising prospects, independently operating LLMs still face challenges such as hallucinations and the lack of verifiable evidence support, necessitating more robust frameworks.

2.2. LLM-based Multi-Agent RAG

RAG has become a pivotal technology for enhancing the capabilities of LLMs in knowledge-intensive tasks [4],[5],[14]. By retrieving relevant external information to ground generation, RAG effectively mitigates the "knowledge cutoff" and hallucination issues of frozen models. In the context of email security, recent work has explored utilizing RAG to retrieve historical email patterns or threat intelligence reports to assist classification. For instance, Zhang et al. [1]systematically reviewed the applications of LLMs in cybersecurity. Similarly, Sniegowski et al. [15]combined RAG with few-shot learning, enhancing the detection capability for rare types of social engineering attacks.

However, most existing RAG-based email classifiers rely solely on a single retrieval strategy (sparse or dense) and often treat retrieval as a static step [16]. Our work advances this direction by introducing multi-agent collaboration into the retrieval process, employing a dual-path retrieval architecture managed by a dedicated RAG Agent that supports both sparse (TF-IDF) and dense (Embedding) modes. This ensures that the retrieved contexts are not only accessed but also undergo rigorous relevance evaluation before influencing the final decision.

ISSN: 2664-9640

2.3. LLM-based Multi-Agent Collaborative Systems

The field of artificial intelligence is experiencing a paradigm shift from single-model reasoning to collaborative Multi-Agent Systems (MAS). Frameworks such as ChatDev [17]and MetaGPT [18]demonstrate that decomposing complex tasks into subtasks handled by dedicated agents (e.g., programmers, reviewers, testers) can significantly enhance performance and robustness. In the cybersecurity domain, Castro et al. [2]introduced an Agentic AI-based cyber defense framework, achieving high-accuracy threat detection through multiple collaborative agents (text analysis, URL scanning, metadata verification, etc.). Despite these advancements, many existing multi-agent security frameworks still focus on division of labor at the feature level rather than collaboration at the cognitive level. In contrast, the hierarchical collaboration of our ARC-Guard framework enables a deeper semantic understanding of attacks compared to simple ensemble methods, as evidenced by our excellent performance on SecEmail.

3. Methodology

3.1. Framework Architecture Overview

We propose ARC-Guard, a collaborative multi-agent framework designed to simulate the cognitive process of security analysts. Unlike traditional monolithic models, ARC-Guard decouples the complex email classification task into three sequential and interdependent stages, each executed by a dedicated agent to ensure comprehensive coverage from surface-level feature extraction to deep logical reasoning.

First, the Initial Analysis Agent acts as the entry point of the system, conducting a rapid scan of the raw email. It does not draw direct conclusions but focuses on extracting key structured features (e.g., sender anomalies, urgent tone) to provide a focused analytical foundation for subsequent steps.

Subsequently, the Dual-Path RAG Agent introduces external knowledge for context augmentation. Given the knowledge cutoff problem inherent in LLMs, this agent supports sparse (TF-IDF) or dense (Embedding) vector retrieval techniques, enabling the recall of annotated cases most similar to the current email from a historical database based on configurations. This step effectively anchors the current decision on real historical data.

Finally, the Chain-of-Thought Agent serves as the ultimate decision-maker. It receives the raw email, the preliminary analysis report, and the retrieved reference cases as inputs. Leveraging the reasoning capabilities of Large Language Models, this agent performs multidimensional logical deduction, comparing the similarities and differences between the current email and historical cases, thereby generating highly interpretable classification results.

Formally, let x denote the input email text, and $y \in Ham, Phish, Spam$ denote the target classification label. The overall reasoning process can be formulated as a composite function:

$$y = F_{\text{CoT}}\left(F_{\text{RAG}}\left(F_{\text{Analysis}}(x)\right)\right)$$

where each function F corresponds to the operation of a specific agent.

3.2. Initial Analysis Agent

As the preprocessing module of the entire classification pipeline, the core task of the Initial Analysis Agent is to perform deep feature perception and deconstruction on the unstructured raw email text x, thereby providing multidimensional semantic features for subsequent decisions. Unlike traditional rule-based systems that rely solely on static feature matching, this

ISSN: 2664-9640

agent utilizes the powerful context understanding capabilities of Large Language Models to deeply parse the social engineering intent behind the text.

Specifically, this agent executes a multidimensional deep scanning process. First, it performs identity consistency verification, checking not only the surface-level match between the sender's address and display name but also identifying domain obfuscation and forged organizational affiliations through semantic analysis. Second, the agent conducts psychological manipulation detection, capturing implicit urgency, intimidation, or alluring promises in the text via sentiment analysis and tone recognition techniques, which are often key tactics used by attackers to bypass rational defenses. Finally, it is responsible for delivery vector risk assessment, deeply scanning embedded URL structures and attachment types within the email, and judging the rationality of their presence in combination with the context, effectively distinguishing normal business links from potential phishing traps. The multidimensional Contextual Analysis A generated by this process transforms the raw text into a set of interpretable risk indicators, substantially reducing the search space in subsequent reasoning stages.

To realize this process, we designed a dedicated prompt template P_{analysis}. The agent first transforms the input text into an analytical context A containing key features, which explicitly points out suspicious points or normal characteristics in the email, providing focused prior information for subsequent reasoning. This feature extraction process can be formalized as maximizing the generation probability from the input text x to the analytical text A under the given prompt P_{analysis}:

$$A = \arg\max_{A'} P\left(A' \mid x, P_{\text{analysis}}\right)$$

3.3. Dual-Path RAG Agent

The Dual-Path RAG Agent serves as the knowledge augmentation module of the system, aiming to resolve the knowledge cutoff and hallucination issues that LLMs may encounter when processing novel attacks or domain-specific terminology. By introducing an external knowledge base D, this agent transforms closed model reasoning into an open, evidence-driven process, ensuring that decisions are grounded in real and relevant historical data.

Considering the diverse feature distribution exhibited by email threats, we design a dual-path retrieval strategy to explore the effectiveness of different feature recall mechanisms.

(1) Sparse Retrieval (TF-IDF):

Sparse retrieval focuses on capturing precise keyword matches. We adopt the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm to construct a vector space containing 5,000 high-frequency feature words. This approach is highly sensitive to identifying known phishing signatures, technical jargon, and specific spam keywords.

For a query email q and a document d in the corpus, the similarity score is calculated as the cosine similarity between their TF-IDF vectors v_q and v_d:

$$\text{Score}_{\text{sparse}}(q,d) = \cos\left(v_q, v_d\right) = \frac{v_q \cdot v_d}{|v_q||v_d|}$$

where the vector weights are defined by $w_{t,d} = \text{tf}(t,d) \times \text{idf}(t)$.

(2) Dense Retrieval (Embedding):

Dense retrieval focuses on capturing deep semantic intent. We utilize deep neural networks to map emails into a high-dimensional continuous vector space $\mathbb{R}^{\mathbb{k}}$. This enables the system to identify complex variant attacks that have undergone adversarial modification in wording but

ISSN: 2664-9640

share core intents (e.g., inducing fund transfers, credential theft) similar to historical cases. The semantic similarity is defined as:

$$\text{Score}_{\text{dense}}(q,d) = \frac{E(q) \cdot E(d)}{|\text{E}(q)||\text{E}(d)|}$$

where $E(\cdot)$ denotes the embedding function.

In practical operation, the RAG Agent activates a specific retrieval path based on configuration, selecting the top-k scoring examples ($k = 3$ in this study) to form the retrieved context set C. Let the retrieval function be $R(q, D)$, where q is the query and D is the knowledge base; the retrieval process can be represented as:

$$C = \{(x_i, y_i) \mid (x_i, y_i) \in \text{Top-}k\big(R(x, D)\big)\}$$

where the similarity scoring function $S(x, x_i)$ is defined according to the different retrieval strategies:

$$S(x, x_i) = \begin{cases} TF - IDF(x) \cdot TF - IDF(x_i) if Sparse \\ \cos(\, Emb(x), Emb(x_i)) \qquad if\ Dense \end{cases}$$

This design allows the system to perform targeted context augmentation for different types of threat features, while also providing an experimental basis for evaluating the respective advantages of sparse and dense retrieval in the security domain.

3.4. Chain-of-Thought Agent

The Chain-of-Thought Agent is the decision-execution core of ARC-Guard. Unlike standard classifiers that perform a black-box mapping directly from text to labels, this agent leverages the powerful reasoning capabilities of Large Language Models to execute explicit, step-by-step logical deduction, thereby endowing the system with high interpretability and robustness. The input to this agent aggregates multi-source information from preceding stages, including the raw email text x, the structured initial analysis report A, and the reference case set C retrieved via RAG.

To guide the model in deep reasoning, we designed a dedicated "Deep Thinking" mode prompt P_{deep}, forcing the model to follow a rigorous cognitive closed-loop. First, the model enters the feature verification and conflict detection phase, where it cross-references the suspicious features marked in the initial analysis report with the retrieved historical cases. For example, if the initial analysis flags a certain urgent tone, and a similar business prompting pattern also exists in retrieved legitimate emails, the model will re-evaluate the maliciousness of this feature. Subsequently, the model executes logical deduction and intent reconstruction, synthesizing all observed evidence to attempt to restore the sender's true intent. This process is no longer confined to surface-level keyword matching but delves into the semantic level, analyzing whether the sender's behavioral patterns conform to their claimed identity.

After completing the internal logic construction, the model conducts comparative analysis and pattern matching, performing meticulous feature alignment between the current email and the retrieved positive and negative samples. By identifying the isomorphism in narrative structure and inductive methods between the current email and known phishing attacks, or the differences in business processes from normal emails, the model can obtain strong discriminative evidence. Finally, based on the cumulative confidence of all the aforementioned

ISSN: 2664-9640

reasoning steps, the agent generates the ultimate verdict and outputs a decision explanation containing the complete chain of thought.

Mathematically, the generation process of the final label y can be modeled as a posterior probability maximization problem under the given context and reasoning path. Let Z denote the intermediate reasoning text generated during the chain-of-thought reasoning process; the final decision process can be decomposed as:

$$y = \arg\max_{c\in\{0,1,2\}} \sum_{Z} P(c \mid Z, C)P\left(Z \mid x, A, C, P_{\text{deep}}\right)$$

This explicit reasoning chain design not only improves classification accuracy (by forcing the model to deliberate) but also endows the system with strong interpretability, ensuring that every security decision is well-documented and traceable.

4. Experiments

4.1. Dataset and Knowledge Base

To rigorously evaluate the effectiveness of ARC-Guard, we constructed a specialized benchmark dataset, SecEmail. We utilized a dataset containing over 300,000 email samples. Through rigorous data cleaning, deduplication, and annotation processes, we filtered out 1,200 high-quality unique emails, ensuring a balanced distribution of Ham, Phishing, and Spam categories.

Subsequently, we randomly split this curated dataset at an 80:20 ratio: 960 emails were designated as the RAG corpus (knowledge base) to provide rich historical precedents for the RAG Agent, while the remaining 240 emails served as the test set to evaluate the generalization capability of the system.

4.2. Experimental Design

To systematically dissect the contribution of each module in the ARC-Guard framework to the overall performance, we designed a multidimensional ablation experimental matrix covering three key dimensions: backbone model selection, reasoning mode differences, and retrieval strategy evolution.

First, regarding the selection of backbone models, we adopted Qwen-Plus [19]and DeepSeek-V3 [20]as the foundation. Both models represent the state-of-the-art in current open-source and closed-source domains, possessing strong instruction-following capabilities and long context windows, which can effectively support complex security analysis tasks.

Second, addressing the impact of Chain-of-Thought (CoT) on decision quality, we compared two different reasoning strategies to establish a performance reference. Direct Reasoning simulates traditional zero-shot classification scenarios, merely requiring the model to output labels directly based on internal knowledge, serving as the logical baseline for measuring reasoning gains. In the ARC-Guard framework, this mode effectively represents scenarios where the "Initial Analysis" and "Chain-of-Thought" agents are inactive or weakened. In contrast, the Deep Thinking Mode fully activates the core cognitive pathway of the system, achieving deep cognitive synergy among the three agents by forcing the model to execute an explicit chain of thought: "feature verification-logical deduction-comparative analysis-final verdict."

Finally, to validate the effectiveness of Retrieval-Augmented Generation (RAG), we constructed three progressive retrieval configurations. The No-RAG (Baseline) setting relies solely on the internal parametric knowledge of the model, used to establish the performance lower bound. Sparse RAG (TF-IDF) introduces a retrieval mechanism based on keyword matching, aiming to capture historical cases containing specific technical terminology or malicious signatures.

ISSN: 2664-9640

Meanwhile, Dense RAG (Embedding) utilizes vector embedding techniques to capture deep semantic similarities, addressing complex attacks with variable wording but identical intent. Through cross-validation of these 12 experimental configurations, we aim to comprehensively reveal the synergistic effects among different modules and the best practices for email classification tasks.

4.3. Results Analysis

The experimental results in Table 1 clearly demonstrate the impact of different modules on the performance of the ARC-Guard framework. We provide in-depth analyses from the following perspectives:

Table 1. Performance Comparison Under Different Experimental Configurations

Model	Reasoning Strategy	RAG Strategy	Accuracy	Precision	Recall	F1 Score
Qwen	Direct Reasoning	No RAG	0.4875	0.5956	0.4875	0.4525
		Sparse RAG	0.7208	0.8096	0.7208	0.6721
		Dense RAG	0.7125	0.8425	0.7125	0.6619
	CoT	No RAG	0.5500	0.6302	0.5500	0.5527
		Sparse RAG	0.9042	0.9065	0.9042	0.9045
		Dense RAG	0.8875	0.8918	0.8875	0.8880
DeepSeek	Direct Reasoning	No RAG	0.5625	0.5628	0.5625	0.5179
		Sparse RAG	0.7000	0.8390	0.7000	0.6364
		Dense RAG	0.7125	0.8397	0.7125	0.6531
	CoT	No RAG	0.5750	0.5450	0.5750	0.5187
		Sparse RAG	0.8542	0.8650	0.8542	0.8504
		Dense RAG	0.8500	0.8772	0.8500	0.8449

Effectiveness of Retrieval-Augmented Generation. The experimental data strongly substantiate the critical role of the RAG mechanism in mitigating hallucinations in Large Language Models. Compared to the baseline model relying solely on internal parametric knowledge (No-RAG), the introduction of the retrieval mechanism yields a significant performance leap. Taking the Qwen model as an example, under Direct Reasoning, the transition from No-RAG to Sparse RAG substantially increases accuracy from 48.75% to 72.08%, a relative increase of nearly 48%. This result validates that the external knowledge base can effectively fill the model's "knowledge gaps." Particularly when handling cases involving specific domains, industry terminology, or novel attack vectors, the contextual anchors provided by RAG play a decisive corrective role.

Advantages of Chain-of-Thought Reasoning. Analysis reveals that CoT reasoning does not function in isolation but forms strong synergistic effects with high-quality retrieved contexts. Across all experimental configurations, models incorporating the Deep Thinking mode achieved higher F1 scores than their Direct Reasoning counterparts under the same conditions. Most notably, the Qwen-Deep-SparseRAG configuration achieved a SOTA accuracy of 90.42%. This breakthrough achievement is attributed not only to the logical rigor brought by CoT but also to the precise recall of critical evidence by sparse retrieval. Compared to Direct Reasoning (72.08%), the performance improvement under the Deep Thinking mode indicates that explicit reasoning steps can assist the model in more effectively utilizing fragmented retrieved information to construct a complete chain of evidence. This confirms that explicit reasoning pathways enable the model to better handle complex social engineering attacks.

Performance Discrepancies Between Retrieval Strategies. A noteworthy and counterintuitive finding is that, within the specific domain of email classification, Sparse

ISSN: 2664-9640

Retrieval (TF-IDF) consistently outperforms or performs on par with Dense Retrieval (Embedding). For Qwen under the Deep Thinking mode, Sparse RAG (90.42%) marginally surpasses Dense RAG (88.75%). We delve into this phenomenon and attribute its root cause to the feature distribution of email threats: many phishing emails contain highly distinctive discrete keywords (e.g., specific forged domain suffixes, technical malicious code snippets, or fixed scam scripts). Sparse retrieval astutely captures these "hard features" through precise keyword matching, whereas semantic-based dense retrieval, despite excelling in intent similarity, might smooth out these crucial nuances during the embedding process, thereby leading to slightly inferior precision when handling template-based attacks.

5. Conclusion

This study proposes and evaluates the ARC-Guard framework, a novel multi-agent collaborative email classification method that combines the advantages of RAG and CoT. Validated through extensive experiments on the SecEmail dataset, ARC-Guard significantly outperforms traditional baselines and independent LLMs across multiple performance metrics. In particular, the configuration combining Qwen-Plus, Sparse RAG, and Chain-of-Thought reasoning achieves a SOTA accuracy of 90.42%. Inspired by the cognitive processes of human security analysts, the design of this framework leverages hierarchical collaboration: the Initial Analysis Agent provides perception, the Dual-Path RAG Agent provides memory, and the Chain-of-Thought Agent provides reasoning. This three-tier architecture offers a robust mechanism for distinguishing complex social engineering attacks from legitimate communications. Our work not only advances the field of email security by enhancing accuracy and interpretability but also provides valuable insights for developing responsible information systems in vertical domains. By achieving more reliable and transparent content evaluation, ARC-Guard facilitates the creation of automated security systems, thereby better supporting SOC analysts in mitigating increasingly complex digital threats.

References

[1] Zhang J, Bu H, Wen H, et al. When llms meet cybersecurity: A systematic literature review[J]. Cybersecurity, 2025, 8(1): 55.

[2] Lazer S J, Aryal K, Gupta M, et al. A Survey of Agentic AI and Cybersecurity: Challenges, Opportunities and Use-case Prototypes[J]. arXiv preprint arXiv:2601.05293, 2026.

[3] Shinn N, Cassano F, Gopinath A, et al. Reflexion: Language agents with verbal reinforcement learning[J]. Advances in neural information processing systems, 2023, 36: 8634-8652.

[4] Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: A survey[J]. arXiv preprint arXiv:2312.10997, 2023, 2(1): 32.

[5] Sharma C. Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers[J]. arXiv preprint arXiv:2506.00054, 2025.

[6] Altwaijry N, Al-Turaiki I, Alotaibi R, et al. Advancing phishing email detection: A comparative study of deep learning models[J]. Sensors, 2024, 24(7): 2077.

[7] Kyaw P H, Gutierrez J, Ghobakhlou A. A systematic review of deep learning techniques for phishing email detection[J]. Electronics, 2024, 13(19): 3823.

[8] He D, Lv X, Xu X, et al. Double-layer detection of internal threat in enterprise systems based on deep learning[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 4741-4751.

[9] Hosseinzadeh M, Ali U, Ali S, et al. Improving phishing email detection performance through deep learning with adaptive optimization[J]. Scientific Reports, 2025, 15(1): 36724.

[10] Tang R, Chuang Y N, Hu X. The science of detecting LLM-generated text[J]. Communications of the ACM, 2024, 67(4): 50-59.

ISSN: 2664-9640

[11] Koide T, Fukushi N, Nakano H, et al. Chatspamdetector: Leveraging large language models for effective phishing email detection[C]//International Conference on Security and Privacy in Communication Systems. Cham: Springer Nature Switzerland, 2024: 297-319.

[12] Heiding F, Schneier B, Vishwanath A, et al. Devising and detecting phishing emails using large language models[J]. IEEE Access, 2024, 12: 42131-42146.

[13] Goldenits G, Koenig P, Raubitzek S, et al. Small Language Models for Phishing Website Detection: Cost, Performance, and Privacy Trade-Offs[J]. arXiv preprint arXiv:2511.15434, 2025.

[14] Yu Y, Ping W, Liu Z, et al. Rankrag: Unifying context ranking with retrieval-augmented generation in llms[J]. Advances in Neural Information Processing Systems, 2024, 37: 121156-121184.

[15] Nilsson P. Phishing for Trust in the AI Age: A Quasi-Experimental Study on Individual Human Factors Influencing Trust in AI-Driven Phishing Attempts[J]. 2024.

[16] Edge D, Trinh H, Cheng N, et al. From local to global: A graph rag approach to query-focused summarization[J]. arXiv preprint arXiv:2404.16130, 2024.

[17] Qian C, Liu W, Liu H, et al. Chatdev: Communicative agents for software development[C]//Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers). 2024: 15174-15186.

[18] Hong S, Zhuge M, Chen J, et al. MetaGPT: Meta programming for a multi-agent collaborative framework[C]//The twelfth international conference on learning representations. 2023.

[19] Bai J, Bai S, Chu Y, et al. Qwen technical report[J]. arXiv preprint arXiv:2309.16609, 2023.

[20] Bi X, Chen D, Chen G, et al. Deepseek llm: Scaling open-source language models with longtermism[J]. arXiv preprint arXiv:2401.02954, 2024.