SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Large Language Model Integration in Enterprise Financial Platforms: Challenges, Patterns, and Emerging Opportunities for Intelligent Customer Engagement

Dinesh Reddy Kasu [1*]

[1*]Independent Researcher, USA

Abstract

Background: Large language models (LLMs) have rapidly transitioned from research curiosities to production components in enterprise financial platforms. Objective: This article examines the architectural and operational challenges of LLM integration in enterprise financial platforms, with particular attention to retrieval-augmented generation (RAG) for domain-specific knowledge grounding, security and data privacy controls, output reliability and hallucination mitigation, and the organizational governance structures required to manage LLM-powered capabilities responsibly. Methods: The study applies a systematic review of architectural patterns drawn from large-scale deployments and synthesizes engineering principles applicable across retail, digital banking, and financial advisory contexts. Results: Four principal engineering challenge domains are identified: knowledge grounding via RAG, security and privacy enforcement, hallucination mitigation through structured validation and escalation, and governance through model lifecycle management and regulatory alignment. Conclusion: Financial institutions that invest in these architectural and governance foundations are positioned to deploy LLM capabilities at scale with the reliability, security, and compliance standards their customers and regulators expect.

Keywords: Large language models, retrieval-augmented generation, enterprise financial platforms, hallucination mitigation, prompt injection, conversational AI, model governance.

INTRODUCTION

The capabilities introduced by large language models (LLMs) represent a qualitative shift in what is achievable in financial services artificial intelligence (AI). Earlier natural language processing (NLP) systems relied on explicit intent classification taxonomies, hand-crafted entity extractors, and template-based response generation, all of which required substantial ongoing engineering effort to maintain and extend [1]. LLMs trained on broad corpora can handle linguistic variation, implicit context, and compositional queries with a fluency that rule-based systems cannot approach. For financial institutions, this capability translates into virtual assistants that can engage in substantive multi-turn conversations about complex topics, advisory tools that synthesize information from multiple sources to answer nuanced client questions, and document processing systems that extract and reason about unstructured financial data [2]. Yet the characteristics that make LLMs powerful also make them challenging to integrate into financial services contexts. LLMs are probabilistic models that can generate syntactically plausible but factually incorrect or nonsensical text, known as hallucinations [3]. In finance, this is particularly problematic since incorrect information could materially affect customers. Customary LLMs trained on general data are not aware of proprietary financial products and regulatory and institutional requirements, preventing them from being used reliably in customer support channels of financial institutions [4]. LLMs could also introduce new vulnerabilities to financial services channels, such as prompt injection, where adversarial prompts in customer messages override instruction prompts to the model [5]. Addressing these challenges requires architectural patterns and governance structures that go well beyond standard software engineering practice. Despite growing industrial deployment of LLM-powered financial applications, the academic literature has focused predominantly on model-level concerns, pre-training, fine-tuning, and benchmark evaluation with comparatively limited treatment of the systems engineering challenges encountered in regulated production environments [6]. This article addresses that gap. The research gap motivating this work is the absence of a consolidated engineering framework that bridges LLM capabilities and the specific operational constraints of regulated financial environments. Prior work by Lewis et al. [7] introduced retrieval-augmented generation as a mechanism for

knowledge grounding, and subsequent literature has advanced prompt engineering and output evaluation techniques [8], but systematic treatment of how these components integrate within enterprise financial architectures, including security controls, escalation design, and regulatory governance, remains underdeveloped. The contributions of this article are as follows: (1) a structured characterization of the principal engineering challenges of LLM integration in enterprise financial platforms; (2) an architectural analysis of retrieval-augmented generation pipelines adapted for financial knowledge bases, including access control and security requirements; (3) a taxonomy of hallucination mitigation strategies with applicability guidance for financial use cases; and (4) a governance framework for model lifecycle management and regulatory alignment. The discussion draws on patterns emerging from large-scale deployments serving retail investors, digital banking customers, and financial advisors. The remainder of this article is organized as follows. Section 2 reviews related work on LLM integration, RAG, and AI governance in financial services. Section 3 describes the methodology. Section 4 presents results organized around the four principal challenge domains. Section 5 discusses the findings, limitations, and directions for future work. Section 6 concludes the article.

LITERATURE REVIEW

Large Language Models and Their Financial Applications

LLM integration into enterprise applications has been studied in the fields of NLP, software systems engineering, AI safety, and financial informatics. The following sections summarize the foundational works most closely related to the engineering challenges addressed in this article. Bommasani et al. [2] provided a comprehensive characterization of foundation models, large models trained on broad data and adapted to diverse downstream tasks, including an extensive treatment of their social and technical risks. Financial NLP research predating the LLM era established domain-specific benchmarks, entity recognition systems, and sentiment analysis tools tailored to financial text [11]. The FinBERT model demonstrated the value of domain-adaptive pre-training for financial text classification [12]. More recent work has explored the application of general-purpose LLMs to financial question answering, earnings call analysis, and regulatory document understanding [13], with results generally showing strong baseline performance but persistent failure modes on domain-specific factual queries, precisely the hallucination problem addressed in this article.

Retrieval-augmented generation

Retrieval-augmented generation (RAG) was formally introduced by Lewis et al. [7], who demonstrated that augmenting language model inputs with retrieved passages from a non-parametric memory store significantly improved performance on knowledge-intensive NLP tasks. The RAG architecture decouples the storage of factual knowledge from the language generation process, enabling dynamic knowledge updates without model retraining. Subsequent work has extended the RAG paradigm to multi-step retrieval [14], structured knowledge sources [15], and domain-specific corpora, with financial services representing an emerging application domain of particular interest given the requirement for factual precision. In financial applications, RAG addresses two complementary requirements: grounding model outputs in authoritative product and regulatory information and providing access to customer account data under appropriate authorization constraints. The latter requirement introduces access control complexities not present in the open-domain retrieval settings studied in most academic RAG literature [16].

Security and Prompt Injection

The security of LLM-powered applications has attracted significant research attention since Perez and Ribeiro [5] systematically characterized prompt injection attack techniques. Subsequent work has demonstrated that prompt injection represents a practical attack vector in deployed applications, capable of causing unauthorized data disclosure, instruction override, and behavioral manipulation [17]. Financial applications are particularly high-value targets for such attacks given the sensitivity of the data they process and the regulatory consequences of unauthorized disclosure [18]. Mitigation approaches studied in the literature include prompt hardening through adversarial training, input sanitization pipelines, and output monitoring classifiers [19]. However, no currently available mitigation provides comprehensive protection against prompt injection, and the literature consistently recommends defense-in-depth strategies combining multiple mitigation layers [20].

Hallucination and Output Reliability

Hallucination in LLMs, the generation of factually incorrect outputs that are nonetheless syntactically fluent and contextually plausible, has been extensively characterized [3]. Zhang et al. [3] divide hallucinations into categories of intrinsic hallucinations, where the model produces inconsistent information regarding the source, and extrinsic hallucinations, where the model produces information that is absent from all sources. They categorize proposed mitigation strategies into training time, decoding time, and post-hoc verification strategies. Structured output

validation that constrains model outputs to schemas verifiable against authoritative data has been proposed as a particularly suitable mitigation for high-stakes factual queries in enterprise environments [21].

AI Governance and Regulatory Alignment

Issues of AI system governance for regulated sectors have been taken up by academics and standards authorities, with the NIST AI RMF 1.0 [22] providing a framework for the identification, assessment, and management of risk as AI models progress through their lifecycle. The Federal Reserve's SR 11-7 guidance on model risk management [23], while predating contemporary LLMs, establishes foundational principles for model validation, documentation, and oversight that apply directly to LLM deployments. Academic work on AI governance frameworks has emphasized the challenges of applying these principles to probabilistic generative models whose input-output relationships are not transparently interpretable [24].

METHODOLOGY

This article employs a systematic architectural review methodology, drawing on published research, industry technical literature, regulatory guidance, and documented deployment patterns to synthesize an engineering framework for LLM integration in enterprise financial platforms. The methodology is appropriate for the research objective, which is to characterize and structure a domain of engineering practice rather than to evaluate a specific technical intervention through controlled experimentation.

The review process involved three stages. First, a targeted literature search was conducted across IEEE Xplore, ACM Digital Library, arXiv, and Springer Link using the query terms "large language models," "retrieval-augmented generation," "financial NLP," "prompt injection," "hallucination," and "AI governance," restricted to publications from 2018 to 2024 to ensure relevance to current LLM technology. Sources were selected based on citation count, venue prestige, and direct relevance to the engineering challenges under examination.

Second, regulatory guidance documents from the Federal Reserve, NIST, and the Financial Industry Regulatory Authority (FINRA) were reviewed to establish the compliance requirements that architectural patterns must satisfy. These documents provide authoritative specifications of the governance and oversight expectations to which deployed systems are accountable.

Third, the identified architectural patterns and governance requirements were synthesized into a structured framework organized around the four principal challenge domains. The framework is evaluated against the dual criteria of engineering feasibility, that the described approaches correspond to implementable system designs, and regulatory alignment, so that the governance structures satisfy documented regulatory expectations.

The methodology is limited in that it does not include primary data collection from specific deployed systems. Production LLM deployments in financial institutions are generally not documented in public-access literature, and institutional data-sharing constraints preclude direct empirical study. The framework, therefore, draws on patterns that are common across documented deployments rather than on measurements from any specific implementation. This limitation is addressed in the discussion section.

RESULTS

The systematic review identified four principal engineering challenge domains for LLM integration in enterprise financial platforms: (1) knowledge grounding through retrieval-augmented generation, (2) security and data privacy enforcement, (3) output reliability and hallucination mitigation, and (4) organizational governance and regulatory alignment. The following sections present the findings for each domain.

Retrieval-Augmented Generation for Financial Knowledge Grounding

Retrieval-augmented generation approaches solve the problem of knowledge grounding by passing relevant institution-specific context to the LLM at inference time instead of relying on what is encoded in model weights during pretraining. A typical RAG architecture consists of a retriever that queries a knowledge base of institution-specific documents using the customer query as a search signal. It identifies relevant documents or passages " which are then injected into the LLM prompt context. The LLM uses this information, along with its general reasoning ability, to produce fluent outputs grounded in accurate institutional knowledge [7] .

The RAG system's knowledge base is controlled, versioned, and up-to-date with the latest product information, regulatory rules and guidance, and internal firm policies. Such content may include product prospectuses, fee schedules, compliance disclosures, product eligibility and exemptions, limits and restrictions, and terms of service,

depending on the financial services line of business. When building a retrieval system, it is always better to optimize for precision rather than recall since we want the retriever to return the most relevant specific documents and not ones that could confuse the model. Embedding-based semantic search, which retrieves documents based on semantic similarity rather than keyword matching, generally outperforms keyword-based retrieval for the natural language queries typical in financial customer interactions [15]. Table 1 presents the principal components of an RAG pipeline adapted for enterprise financial applications, together with the financial-services-specific considerations applicable to each component.

Table 1 Components of a retrieval-augmented generation pipeline for enterprise financial applications

Pipeline Component	Function	Financial Services Consideration
Knowledge base	Stores curated institutional documents	Must be versioned and access-controlled
Embedding model	Converts text to semantic vectors	Must be evaluated for domain-specific financial vocabulary
Retrieval engine	Finds the most relevant documents per query	Precision, critical wrong documents mislead the model
Permissions layer	Filters documents by user authorization	Prevents non-public data exposure to customers
Prompt builder	Injects retrieved context into the model prompt	Must be hardened to resist prompt injection attacks
Output validator	Verifies response against authoritative data	Required for regulatory accuracy compliance

A critical design decision in financial RAG systems is the granularity of the knowledge base. Chunking documents at the paragraph level is generally seen as superior to document-level retrieval. The model only needs to see the passage that answers their customer query rather than reading through the entire document or product prospectus. Chunking the documents too much makes it difficult to remember earlier information to answer questions with multiple parts. Empirical evaluation of chunking strategies against a representative sample of production queries is, therefore, an essential step in RAG pipeline development for financial applications.

Security and Data Privacy Controls

Financial knowledge bases contain a mixture of public product information and non-public institutional data. RAG systems must ensure strict access control to prevent the retrieval or injection of any non-public data into prompts that are processed with consumer or third-party systems. For this purpose, a permissions layer can be integrated into the retrieval pipeline that filters the retrieved documents based on permissions of the requester [16]. Customers interacting through self-service channels should access only public product information and their own account data, while financial advisors or internal tools may access a broader range of institutional research and proprietary analytics.

Prompt injection represents a distinct and particularly serious security threat in RAG-enabled financial applications [5]. Malicious prompts could also be delivered by users to encode system-level instructions that override the instructions in the LLM prompt and enable the model to leak confidential information, generate inaccurate financial advice, or carry out unauthorized transactions. In RAG systems, adversarial prompt injection attacks could be executed in the documents retrievable from the knowledge base via indirect prompt injection, allowing the model to execute adversarial system-level instructions when the documents are retrieved [17]

Mitigation strategies include prompt hardening techniques that structure system prompts to resist injection, input sanitization that filters potentially adversarial patterns from customer inputs before they reach the model, and output monitoring that scans model-generated responses for signs of prompt injection success before they are delivered to customers [19]. Security testing of LLM-integrated systems must specifically include adversarial injection testing as a standard component of the quality assurance process. The layered nature of effective prompt injection defense no single mitigation is sufficient and requires explicit design decisions about the combination of controls deployed at each layer of the inference pipeline.

Data residency and regulatory data handling requirements impose additional constraints on LLM inference infrastructure in financial services. In jurisdictions with data localization requirements, the inference pipeline, including any external model APIs, must be configured to ensure that customer data does not transit or reside in

unauthorized jurisdictions. This constraint may favor on-premises model deployment or the use of regional cloud infrastructure over generic public cloud API endpoints for customer-facing financial applications.

Output Reliability and Hallucination Mitigation

Hallucination, the generation of factually incorrect but superficially plausible outputs, poses a particularly serious risk in financial services contexts [3]. If hallucinations occur regarding product terms, fees, eligibility criteria, or regulations, there is a risk of customers making financial commitments under pretenses, which can lead to regulatory actions and reputation risk. Reducing the risk of hallucination may require complementary approaches at several layers of LLM integration architecture.

Structured output validation is a method that validates LLM outputs against a schema that is checked against authoritative data sources before being communicated [21]. In this approach, instead of generating a free-text response based on factual information, the model is instructed to generate outputs in a structured format, such as a JSON object with defined fields. The generated outputs are then checked against a knowledge base. For example, for any question about fund expense ratios, the expense ratio field of the structured output will be directly populated from an authoritative fund data store (instead of the model's answer) to prevent the model from generating incorrect values and trade off some conversational fluency for much higher factuality for information where the stakes (e.g., financial information) are particularly high.

Another mitigation applied to ambiguous or out-of-distribution queries is confidence thresholding, where if the internal confidence of the model's generated response falls below a certain threshold, the query is instead escalated to a human agent. This is favored instead of responding with an automatically generated response, which may be inaccurate or factually incorrect. Calibration thresholds should be tuned against empirical model outputs with respect to ground truth using the query distribution available and should be re-evaluated whenever the query distribution or base model changes. Table 2 contains the list of major hallucination mitigation techniques discussed in the review, along with their descriptions and applications.

Table 2 Hallucination mitigation strategies for llm deployments in regulated financial environments.

Mitigation Strategy	Description	Best Applied To
Retrieval-augmented generation	Grounds responses in curated institutional knowledge	Product and policy queries
Structured output validation	Constrains model output to a schema verified against the data store	Factual data queries (rates, fees, eligibility)
Confidence thresholding	Escalates when model confidence falls below a defined threshold	Complex or ambiguous queries
Human-in-the-loop escalation	Routes high-stakes interactions to licensed advisors	Investment advice, estate planning
Output monitoring	Scans responses for inaccuracies before delivery	All customer-facing interactions
Adversarial testing	Tests model robustness to edge cases and injection attempts	Pre-deployment validation

Not all financial queries can be safely handled by fully automated LLM-powered systems. Interactions involving complex investment decisions, tax implications, estate planning, or highly personalized financial planning recommendations require a level of contextual judgment and regulatory accountability that current LLM technology cannot reliably provide. Architectures that deploy LLMs in financial services must therefore incorporate explicit escalation pathways that route high-stakes interactions to licensed human advisors or specialists [23]. Designing effective escalation triggers requires identifying the classes of queries and customer contexts that exceed the reliable capability of the automated system. Risk signals that may indicate the need for escalation include queries involving specific dollar amounts above defined thresholds, interactions with customers who have indicated financial distress, requests for personalized investment advice that would require consideration of the customer's complete financial situation, and any interaction in which the customer explicitly requests human assistance. Systems that transparently acknowledge the boundaries of their competence and smoothly transfer customers to human support consistently achieve higher customer satisfaction scores than systems that attempt to handle all queries autonomously, particularly when errors in automated responses are costly or difficult to reverse[25].

Organizational Governance and Regulatory Alignment

The lifecycle of LLMs in financial applications includes model selection, deployment, monitoring, and model replacement. Model selection compares available foundation models to the application's performance, cost, and regulatory compliance requirements. This process can be repeated when new models become available or as the business application's needs change. Data provenance, documentation of training data, bias assessment, and model card disclosures of the model are also likely to play an important role in this process and may be required by institutional policy or new AI regulation [22].

For LLM powered systems, monitoring and alerting should align with the failure modes of language models. As with other software, it's still important to track uptime, latency, and other resource usage metrics, but quality metrics that track semantic understanding, such as the relevance of the generated responses, the accuracy of the statements as compared to a ground truth dataset, and the number of escalation triggers are also valuable. Monitoring for distributional drift, changes in the statistical distribution of model inputs and outputs, may provide an early indicator that performance may not be sustainable if the distribution of customer queries or the model's institutional knowledge base diverges from the training distribution.

The financial services regulators in major jurisdictions are developing regulatory frameworks for the use of AI in customer-facing applications, which essentially require organizations to show that their AI systems provide explainable, non-discriminatory outcomes and that human oversight mechanisms are present for high-stakes or high-impact decisions [24]. LLMs present specific challenges for regulatory compliance because the relationship between their inputs and outputs is not transparently interpretable in the way that rule-based systems are.

Approaches to regulatory alignment for LLM-powered financial applications include documentation of model selection criteria and evaluation results, maintenance of audit logs of all model inputs and outputs, implementation of bias monitoring pipelines that evaluate model outputs across customer demographic segments, and development of plain-language explanations of AI decision processes for customer-facing disclosures. By taking these steps voluntarily rather than waiting for regulation, these institutions are better positioned for the rapidly scalable deployment of LLM capabilities, and can respond to regulatory inquiries.

DISCUSSION

The findings presented in Section 4 collectively support the conclusion that LLM integration in enterprise financial platforms is an engineering discipline requiring specialized architectural patterns and governance structures not present in general-purpose LLM deployment frameworks. Each of the four challenge domains, knowledge grounding, security, hallucination mitigation, and governance, involves requirements that are either unique to financial services or significantly more stringent than in consumer or research contexts.

The centrality of RAG to financial LLM deployment reflects a fundamental mismatch between the knowledge encoded in general-purpose foundation models and the specific, current, proprietary knowledge required for reliable financial customer interactions. While fine-tuning on financial corpora improves domain vocabulary handling [12], it cannot reliably encode the continuously changing product terms, fee structures, and regulatory requirements that characterize financial services. RAG provides a more maintainable architecture by decoupling dynamic institutional knowledge from model weights, though it introduces its own engineering complexities around retrieval quality, access control, and indirect prompt injection risk.

The prompt injection findings underscore that security analysis for LLM-powered financial applications must extend beyond the conventional threat model of software security. In conventional web applications, the principal attack surfaces are authentication bypass, injection attacks in database queries, and network-level vulnerabilities. LLM-powered applications inherit these surfaces and add the novel attack surface of natural language manipulation of model behavior through adversarial inputs. The financial services threat model is particularly acute because the potential payoff for successful prompt injection access to account data, generation of fraudulent transaction authorizations, or disclosure of confidential institutional information is substantially higher than in most other application domains.

The hallucination mitigation findings reveal a tension between conversational fluency and factual reliability that has no universal resolution. Structured output validation and confidence thresholding each limit the conversational naturalness of LLM responses in exchange for improved reliability, while fully autonomous free-form generation maximizes fluency at the cost of factual control. The appropriate balance depends on the specific interaction context: factual data queries about product terms and fees warrant high validation stringency, while

more exploratory educational conversations about financial concepts may tolerate greater generative freedom. Deploying these mitigations selectively based on query classification represents a promising architectural pattern that warrants further empirical study.

The governance findings highlight a significant maturity gap between the AI governance capabilities required for compliant LLM deployment in financial services and the current state of institutional AI governance practice. Most financial institutions have model risk management frameworks established for conventional predictive models, credit scoring, and fraud detection, but these frameworks were not designed to accommodate the probabilistic, generative characteristics of LLMs. Extending model risk management to LLMs requires new approaches to model validation that go beyond the hold-out accuracy metrics appropriate for conventional models, including semantic quality evaluation, adversarial robustness testing, and distributional drift monitoring.

Comparison with Prior Work

The architectural framework developed in this article extends prior work on RAG [7] and prompt injection mitigation [5] by embedding these techniques within a broader systems engineering context specific to financial services. While Lewis et al. [7] and subsequent RAG literature focus on retrieval effectiveness in open-domain settings, this article addresses the access control and security requirements that distinguish financial RAG from open-domain deployment. Similarly, the hallucination mitigation taxonomy extends Zhang et al. [3] by providing application-context guidance tailored to financial use cases, rather than treating mitigation strategies as universally applicable.

The governance framework complements the NIST AI RMF [22] by providing financial-services-specific elaboration of the framework's general principles. The NIST framework provides a vocabulary and structure for AI risk management, but does not address the specific failure modes and regulatory requirements of LLM deployment in financial services. The model lifecycle management practices and monitoring metrics described in Section 4.4 provide a more operationally concrete guidance set within the NIST framework's structure.

Limitations and Threats to Validity

Several limitations of this study warrant acknowledgment. First, the methodology relies on a synthesis of published literature and documented deployment patterns rather than on primary data from specific production deployments. This limits the empirical grounding of the framework and may miss deployment patterns that have not been publicly documented. Future work involving structured interviews with practitioners at financial institutions deploying LLM capabilities would provide a valuable empirical complement to the literature-based synthesis presented here.

Second, the rapidly evolving capabilities of foundation models mean that future model generations may partially address some of the limitations driving architectural choices in this framework. Improved factual reliability, more robust injection resistance, and better-calibrated confidence scores in future LLMs could relax some of the architectural constraints described here. The governance and regulatory alignment framework is more durable, as it derives from regulatory requirements that evolve more slowly than model capabilities.

Third, the regulatory environment for AI in financial services varies significantly across jurisdictions, and the framework described here primarily reflects the North American and European regulatory context. Institutions operating in other regulatory environments should evaluate the applicability of the governance recommendations to their specific jurisdictional requirements.

CONCLUSIONS

This article has developed a structured engineering framework for LLM integration in enterprise financial platforms, organized around four principal challenge domains: knowledge grounding through retrieval-augmented generation, security and data privacy enforcement, output reliability and hallucination mitigation, and organizational governance and regulatory alignment. The framework's contributions include a component-level analysis of financial RAG pipelines with access control requirements, a taxonomy of hallucination mitigation strategies with application-context guidance, and a model lifecycle governance structure aligned with the regulatory expectations of major financial services jurisdictions.

The core finding of the analysis is that LLM integration in financial services is not primarily a model selection or fine-tuning problem, but a systems engineering problem requiring specialized architectural patterns and institutional governance structures. The probabilistic, generative characteristics of LLMs create failure models, hallucination, prompt injection vulnerability, and distributional drift that have no direct analogues in conventional

financial software and require engineering responses not present in general-purpose LLM deployment frameworks.

Financial institutions that invest in building the architectural and governance foundations described in this article are positioned to deploy LLM capabilities at scale with the reliability, security, and compliance standards their customers and regulators expect. The engineering principles presented here provide a durable foundation for responsible AI innovation in financial services as model capabilities continue to advance and regulatory frameworks mature.

Future research should address the empirical validation of the hallucination mitigation taxonomy through controlled studies in financial deployment contexts, the development of LLM-specific extensions to model risk management frameworks, and the characterization of indirect prompt injection risks in financial RAG systems and their mitigations. Practitioner studies examining the organizational change management challenges of LLM deployment in financial institutions would complement the technical engineering analysis presented here.

Author Contributions
[Dinesh Reddy Kasu]: Conceptualization, formulating the research problem and identifying the four principal engineering challenge domains for LLM integration in enterprise financial platforms; Methodology, designing the systematic architectural review approach and selecting the scope of literature and regulatory documents; Formal Analysis, synthesizing findings across retrieval-augmented generation, security, hallucination mitigation, and governance domains; Investigation, reviewing and interpreting published research, deployment patterns, and regulatory frameworks; Writing Original Draft, authoring all sections of the manuscript; Writing Review and Editing, revising and refining the manuscript for clarity, technical accuracy, and compliance with journal requirements; Visualization, designing the tabular frameworks presented in the results section; project administration, managing all aspects of the submission process.

Funding
This research received no external funding.

Conflicts of Interest
The author declares no conflict of interest.

References

1. Tom Young et al., "Recent trends in deep learning-based natural language processing," Arxiv, 2018. https://arxiv.org/abs/1708.02709
2. Rishi Bommasani et al., "On the opportunities and risks of foundation models," arXiv:2108.07258, 2022. https://arxiv.org/abs/2108.07258
3. Yue Zhang et al., "Siren's song in the AI ocean: A survey on hallucination in large language models," arXiv:2309.01219, 2025. https://arxiv.org/abs/2309.01219
4. Dan Hendrycks et al., "Aligning AI with shared human values," arXiv:2008.02275, 2023 https://arxiv.org/abs/2008.02275
5. Fábio Perez, Ian Ribeiro, "Ignore previous prompt: Attack techniques for language models," Arxiv, 2022. https://arxiv.org/abs/2211.09527
6. Wayne Xin Zhao et al., "A survey of large language models," arXiv, 2023. https://arxiv.org/abs/2303.18223
7. Patrick Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," ACM Digital Library, https://dl.acm.org/doi/abs/10.5555/3495724.3496517
8. Jason Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," arXiv, 2023. https://arxiv.org/abs/2201.11903
9. Ashish Vaswan et al., "Attention is all you need," arXiv, 2023. https://arxiv.org/abs/1706.03762
10. Jared Kaplan et al., "Scaling laws for neural language models," arXiv, 2020. https://arxiv.org/abs/2001.08361
11. arXiv, "Evaluating Large Language Models (LLMs) in Financial NLP: A Comparative Study on Financial Report Analysis," https://arxiv.org/html/2507.22936v1#:~:text=Large%20Language%20Models%20(LLMs)%20have,widely%20used%20LLMs%20remain%20underexplored.
12. Dogu Araci, "FinBERT: Financial sentiment analysis with pre-trained language models," arXiv, 2019. https://arxiv.org/abs/1908.10063
13. Jie Huang et al., "Towards reasoning in large language models: A survey," arXiv, 2023. https://arxiv.org/abs/2212.10403

14. Zhihong Shao et al., "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," arXiv, 2023. https://arxiv.org/abs/2305.15294
15. Weijia Shi et al., "REPLUG: Retrieval-augmented black-box language models," arXiv, 2023. https://arxiv.org/abs/2301.12652
16. Bingxiang Chen et al., Integrating Access Control with Retrieval-Augmented Generation: A Proof of Concept for Managing Sensitive Patient Profiles," ACM Digital Library 2025.
17. https://dl.acm.org/doi/abs/10.1145/3672608.3707848.
18. Kai Greshake et al., "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," arXiv, 2023. https://arxiv.org/abs/2302.12173
19. Tom Czaban, "Security and AI in Financial Services: Balancing Innovation with Risk Management," GoodData, 2025. https://www.gooddata.com/blog/security-and-ai-in-financial-services-balancing-innovation-with-risk-management/
20. Yi Liu et al., "Prompt injection attacks and defenses in LLM-integrated applications," arXiv, 2023. https://arxiv.org/abs/2306.05499
21. Eric Wallace, "Concealed data poisoning attacks on NLP models," arXiv, 2021. https://arxiv.org/abs/2010.12563
22. Han Yuan et al., "Navigating the Impact of Structured Output Format on Large Language Models through the Compass of Causal Inference," arXiv, 2025. https://arxiv.org/html/2509.21791v1
23. Gina M. Raimondo, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1,
24. https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf
25. Board of Governors of the Federal Reserve System. "Federal Reserve Board announces approval of application by OceanFirst Financial Corp.?" 2026[Online]. Available:
26. https://www.federalreserve.gov/newsevents/pressreleases/orders20260424a.htm
27. Dr. David Leslie, "Understanding artificial intelligence ethics and safety: A guide for the responsible design and implementation of AI systems in the public sector," 2019. https://www.turing.ac.uk/sites/default/files/2019-08/understanding_artificial_intelligence_ethics_and_safety.pdf
28. Alec Radford et al., "Language models are unsupervised multitask learners," OpenAI Blog.
29. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf