International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Generative AI in Cloud Computing

Applications, Security Challenges, and Future Research Directions

Ruchita Bharat Konde

Mrs. Asmita Mohan Phatangare (Assistant Professor)

Department of Master of Computer Application,
JCEI's Jaihind Institute of Management and Research, Vadgaon Sahani, Pune, India
Email: konderuchita@gmail.com asmitafatangare41@gmail.com

Abstract - Generative Artificial Intelligence (Generative AI) is increasingly delivered through cloud platforms, where scalable infrastructure, managed model services, and application programming interfaces (APIs) lower the barrier to adoption. This integration also changes the security and privacy risk profile of cloud systems. This paper presents a preliminary, literature-based review that aims to identify major applications, characterize security and privacy risks, compare reported mitigation approaches, and highlight unresolved research questions. Peer-reviewed papers, conference papers, preprints, and authoritative guidance were collected through web-based searches and compared by objectives, methods, findings, and limitations. The reviewed evidence shows concrete uses in cloud incident analysis and code assistance, together with recurring risks involving prompt injection, sensitive-data disclosure, retrieval-data leakage, and excessive permissions granted to AI agents. Reported mitigations, such as provenance marking, execution isolation, and least-privilege design, are promising but are generally evaluated against limited attack sets, and none is presented as a complete solution. The review identifies gaps in cloud-specific security evaluation, agent authorization, and joint analysis of security, performance, and cost.

Key Words: generative AI, cloud computing, large language models, cloud security, data privacy, prompt injection.

1. INTRODUCTION

Cloud computing is defined by the U.S. National Institute of Standards and Technology (NIST) as a model for enabling on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned with minimal management effort [1]. Generative Artificial Intelligence (Generative AI) refers to models that produce new content, such as text, code, and images, rather than only classifying or predicting from existing data. Recent progress builds on the Transformer architecture [2] and on Large Language Models (LLMs) that can perform new tasks from a few examples supplied in the prompt [3]. Because training and serving such models require substantial computing and data-management capacity, cloud platforms have become a principal delivery channel; a recent scoping review describes cloud infrastructure for training, deploying, and scaling generative models [4].

In practice, cloud-hosted Generative AI is rarely a model alone. Applications combine a model service with retrieval of private documents [5], external tools, and third-party Application Programming Interfaces (APIs). Reported benefits include assistance with cloud incident diagnosis [6], [7] and with software development [8]. The same architecture, however, processes prompts, documents, and credentials that may be confidential, and because instructions and data share one natural-language channel, untrusted content can influence model behavior [9]. Security, privacy, scalability, and cost must therefore be considered together; for example, the OWASP list of LLM application risks includes unbounded resource consumption alongside injection and data disclosure [10].

Existing surveys examine LLM security and privacy in general [11], [12], [13], and cloud-platform reviews describe service capabilities [4]. Within the sources reviewed here, however, the cloud deployment context, in which model services, retrieval stores, identity systems, and third-party integrations interact, is seldom analyzed as a single system. This matters to cloud service providers, who set the defaults of managed AI services; to organizations, which remain accountable for their data; to developers, who need secure design patterns; and to users, whose information passes through these systems.

This paper is a preliminary, literature-based review with no experimental component. Its objectives are to (i) summarize applications of Generative AI in cloud computing, (ii) classify the associated security and privacy risks, (iii) compare reported mitigation approaches, and (iv) identify research gaps that the reviewed evidence supports. The contributions are a structured overview of applications that separates demonstrated from emerging uses; a layered classification of risks (model, application, and cloud infrastructure); a qualitative comparison of mitigations; and research gaps labeled by the strength of their evidential basis. The scope is limited to text-oriented LLM systems.

2. BACKGROUND AND RELATED CONCEPTS

2.1 Cloud Computing and Shared Responsibility

NIST defines three service models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and four deployment models: private, community, public, and hybrid cloud [1]. Generative AI capabilities are commonly consumed as managed services through APIs: a client sends a prompt, a provider-hosted model returns generated output, and the customer's application mediates data and actions around that call. Under the shared-responsibility principle used by cloud providers, the provider secures the underlying infrastructure, whereas the customer remains responsible for identity configuration, data, and application logic; the exact boundary depends on the service model.

2.2 Generative AI, LLMs, and RAG

Unlike traditional discriminative machine learning, which maps fixed inputs to labels or numerical predictions, generative

models produce open-ended content from natural-language instructions. That flexibility is also a security concern: instructions and data travel through the same channel, so the model cannot reliably separate them [9]. Retrieval-Augmented Generation (RAG) supplements a model with documents retrieved from an external store at query time [5]. It allows use of private or current information without retraining, but it adds a data store whose confidentiality must be protected [14].

2.3 AI Agents and Access Control

AI agents are LLM-driven systems that call tools or APIs to act on a user's behalf; benchmarks such as AgentDojo evaluate them while they execute tools over untrusted data [15]. Confidentiality of prompts and retrieved content depends on conventional controls, namely authentication, authorization under least privilege, and encryption, applied around a component whose behavior is probabilistic. Fig. 1 shows the conceptual architecture used in the remainder of this paper; it is an original diagram prepared from the concepts above.

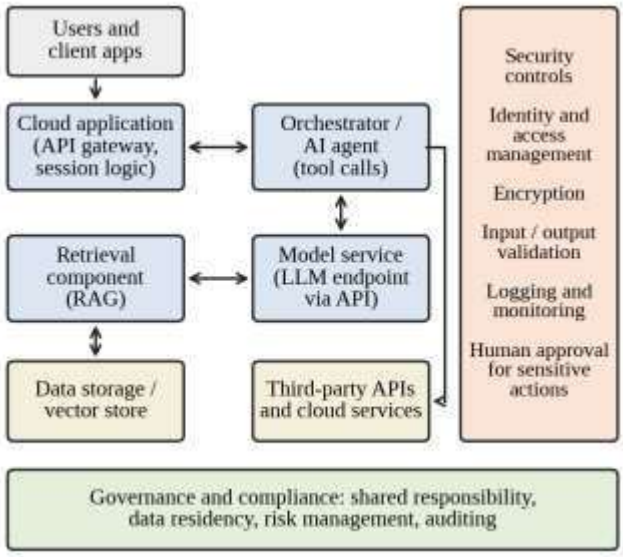


Fig. 1. Conceptual architecture of Generative AI in cloud computing, showing users, cloud application, orchestrator, model service, retrieval, data storage, third-party APIs, and security controls.

3. LITERATURE REVIEW

The reviewed sources fall into four groups, compared in Table 1.

3.1 Generative AI on Cloud Platforms and Operations

Patel *et al.* [4] review services from major providers, covering compute, serverless, edge, and storage, and discuss security, privacy, compliance, vendor lock-in, and sustainability. The work is a preprint and draws partly on provider documentation and industry sources, so its statements on commercial capabilities need cautious reading. Two studies address cloud incident management. Ahmed *et al.* [6] evaluated LLMs for root-causing and mitigating production incidents using more than 40,000 incidents, comparing zero-shot, fine-tuned, and multi-task settings and adding a human evaluation by incident owners. RCACopilot [7] matches incidents to handlers, aggregates diagnostic information, predicts a root-cause category, and generates an explanation; the authors report accuracy of up to 0.766 on a year of incidents and note that its diagnostic-collection component has been in use for over four years. For code assistance, Pearce *et al.* [8] prompted GitHub Copilot with 89 scenarios and found approximately 40% of the 1,689 generated programs to be vulnerable.

3.2 Security and Privacy of LLMs

Yao *et al.* [11] classify work into beneficial uses, offensive uses, and vulnerabilities with defenses, noting that research on model-extraction attacks is limited and often theoretical. Das *et al.* [12] review jailbreaking, data-poisoning, and personal-data-leakage attacks and defenses. Li and Fung [13] summarize 2022–2025 studies on prompt injection, adversarial attacks, misuse, and autonomous-agent risks, and stress layered defenses. These surveys are broad but are not specific to cloud deployment.

3.3 Attacks on LLM-Integrated Applications

Perez and Ribeiro [16] showed that simple handcrafted inputs can hijack the goal of a GPT-3-based application or leak its prompt. Greshake *et al.* [9] introduced indirect prompt injection, in which instructions are planted in data that an application later retrieves, and demonstrated attacks against real systems, including Bing Chat and code-completion engines, as well as GPT-4-based synthetic applications; they report that effective mitigations were lacking. Carlini *et al.* [17] extracted verbatim training sequences, some containing personal information, from GPT-2. Zeng *et al.* [14] showed that RAG systems can leak their private retrieval database while, in their experiments, reducing leakage of training data.

3.4 Defenses, Benchmarks, and Governance

Hines *et al.* [18] propose spotlighting, which transforms untrusted input to signal its provenance, and report that attack success fell from above 50% to below 2% in their experiments with GPT-family models. IsolateGPT [19] isolates the execution of third-party apps and mediates their interaction; the authors report overhead under 30% for three-quarters of tested queries. AgentDojo [15] provides 97 tasks and 629 security test cases and finds that existing attacks break some, but not all, security properties. The OWASP list [10] and the NIST Generative AI Profile [20], a companion to the AI Risk Management Framework [21], offer practitioner-oriented risk taxonomies and management actions rather than experimental evidence.

Comparing the groups, the operational studies [6], [7] use realistic data but come from a single provider and do not evaluate attack resistance. The attack studies establish feasibility but often target model or product versions that change quickly. Defense studies [18], [19] report promising results under defined attack sets, and governance documents [10], [20] are broad but non-experimental.

Table -1: Comparison of Selected Related Studies

Ref.	Objective	Method	Main reported finding	Limitation / relevance
[4]	Review cloud services for generative AI	Scoping review; provider documentation and industry sources	Compares services; notes privacy, compliance, lock-in, and sustainability challenges	Preprint; broad; limited security depth

[6]	LLMs for incident root-causing and mitigation	Over 40,000 incidents; zero-shot, fine-tuned, multi-task; human evaluation	Models evaluated; incident owners' feedback indicates potential	Single provider; security not the focus
[7]	Automate cloud incident root-cause analysis	LLM-based on-call system; one year of incidents	Accuracy up to 0.766; collection component used over four years	Single provider; human validation still relevant
Ref.	**Objective**	**Method**	**Main reported finding**	**Limitation / relevance**
[8]	Security of AI-generated code	89 scenarios, 1,689 programs; static analysis and manual review	Approximately 40% of programs vulnerable	Tool version at time of study
[11]	Map LLM security and privacy	Literature survey	Good/bad/ugly taxonomy; model-extraction research limited	Not cloud-specific
[9]	Indirect prompt injection	Taxonomy; demonstrations on real and synthetic apps	Attacks viable; mitigations lacking at the time	Specific products and models
[14]	Privacy of RAG systems	Empirical attacks on RAG	Retrieval database can leak; training-data leakage reduced	Controlled experiments; not cloud-specific
[15]	Evaluate agent attacks and defenses	Benchmark of 97 tasks, 629 security cases	Agents fail many tasks; attacks break some properties	Benchmark environment, not provider-specific
[18]	Defend against indirect injection	Prompt transformations; GPT-family experiments	Attack success above 50% to below 2% in tests	Tied to tested attacks and models
[19]	Isolate third-party LLM apps	Architecture; case-study attacks	Protects against many tested attacks; overhead under 30% for 75% of queries	Prototype; design-level evidence

4. RESEARCH GAP AND RESEARCH QUESTIONS

Gaps are separated by evidential basis. ***Explicit limitations.*** Greshake *et al.* [9] state that effective mitigations for indirect prompt injection were lacking; Yao *et al.* [11] note that model-extraction research is limited and often theoretical; and Debenedetti *et al.* [15] report that agents fail many tasks even without attacks and call for new design principles.

Patterns across studies. In the sources reviewed, (1) operational cloud studies [6], [7] and security studies [9], [14], [18] rarely overlap: the former do not test attack resistance, and the latter seldom evaluate cloud-specific controls such as identity management or network isolation; (2) defenses are evaluated on particular models and attack sets [18], [19]; and (3) cost and latency are reported for only one of the defenses reviewed [19]. These observations concern a limited corpus and do not prove that the topics are unexplored.

Interpretation. These patterns suggest value in cloud-specific reference architectures and evaluation methods that assess security, performance, and cost together, and in authorization models for agents that go beyond isolation. This interpretation needs validation against a wider literature.

Four research questions guide the review. **RQ1:** What are the major applications of Generative AI in cloud computing environments? **RQ2:** What security and privacy risks arise when Generative AI is deployed through cloud-based services? **RQ3:** What mitigation approaches have been proposed or evaluated, and what limitations remain? **RQ4:** Which research directions could improve the security, privacy, and reliability of cloud-hosted Generative AI? The corresponding objectives are to catalog applications by evidence level (RQ1), classify risks by layer (RQ2), compare mitigations qualitatively (RQ3), and derive research directions (RQ4).

5. RESEARCH METHODOLOGY

Design. This is a preliminary, literature-based narrative review. It is not a systematic review, and no PRISMA-style screening counts are claimed. No experiments were performed.

Sources and search. Searches were run in October 2026 with a general web search tool. The results pointed to publisher, proceedings, preprint, and standards pages, which were used to confirm bibliographic details. Searches were not executed directly inside IEEE Xplore, the ACM Digital Library, SpringerLink, ScienceDirect, or Scopus. The query themes were: generative AI and cloud platforms; LLMs for cloud incident analysis; LLM security and privacy surveys; prompt injection and indirect prompt injection; RAG privacy; LLM agent security and benchmarks; execution isolation for LLM applications; code-assistant security; and NIST and OWASP guidance for generative AI. The publication range was 2017–2026, emphasizing 2022 onward, with earlier foundational works included.

Selection. Inclusion criteria: the source addresses Generative AI or LLMs in cloud or LLM-integrated application contexts, or the security, privacy, or governance of such systems; it is peer-reviewed, a preprint with identifiable authors (flagged as such), or authoritative guidance; and its bibliographic details could be verified on an official page. Exclusion criteria: vendor marketing or blog posts without technical evidence, sources whose details could not be verified, and sources unrelated to the research questions. Relevance was judged from titles, abstracts, and available full-text excerpts. The numbers of records identified and screened were not logged and are therefore not reported; the final list contains 21 sources.

Extraction and analysis. For each source, the objective, method, evidence type, main finding, security challenge addressed, limitation, and relevance were recorded. Sources were classified by layer (model, application, or cloud infrastructure) and evidence type (demonstrated attack, evaluated defense, survey, or guidance). Gaps were identified by comparing recorded limitations and labeled as explicit, pattern-based, or interpretive.

Threats to validity. A web-based search may miss relevant studies, recent work (especially after 2025) may be underrepresented, and selection involved judgment. A systematic search of the databases named above, with logged counts, is recommended before formal submission.

6. APPLICATIONS OF GENERATIVE AI IN CLOUD COMPUTING

Each application is labeled *demonstrated* (empirical or deployed evidence in the reviewed sources) or *emerging* (plausible but not directly evidenced in them).

Cloud operations, troubleshooting, and incident investigation (demonstrated). LLMs receive an incident's title and summary [6] or aggregated diagnostic data [7] and propose root causes, mitigation steps, or explanations. Benefits include reduced manual effort for on-call engineers and reuse of historical incidents. Concerns include the need to verify recommendations, since both studies position the output as support for engineers, and the exposure of operational data in prompts. The evidence comes from a single provider.

Code generation and development assistance (demonstrated). Assistants complete code from context. In the scenarios of Pearce *et al.* [8], a substantial share of generated programs was vulnerable, so productivity gains depend on review and security tooling.

Generative AI delivered through cloud APIs (demonstrated, descriptive). Managed model services reduce the need to operate specialized hardware [4]. The same review lists data

privacy, compliance, model protection, vendor lock-in, and sustainability as open concerns.

Data analysis and knowledge retrieval (demonstrated). RAG lets applications answer from enterprise documents [5], but retrieval stores can themselves leak [14].

AI agents interacting with cloud services (emerging). Agents execute tools over external data. AgentDojo shows that evaluated agents fail many tasks even without attacks and remain open to injected instructions [15], and isolation architectures are being explored [19]. An agent holding broad cloud credentials would increase the consequences of manipulation (reasoned inference).

Resource management, customer support, and monitoring beyond incidents (emerging). No study reviewed here directly evaluates Generative AI for cloud resource management or customer support, so these are treated as plausible extensions of the uses above rather than demonstrated applications.

7. SECURITY AND PRIVACY CHALLENGES

Risks are grouped into model-level, application-level, and cloud-infrastructure-level categories (Fig. 2). Unless a source is cited, statements below are conceptual analysis, not documented incidents.

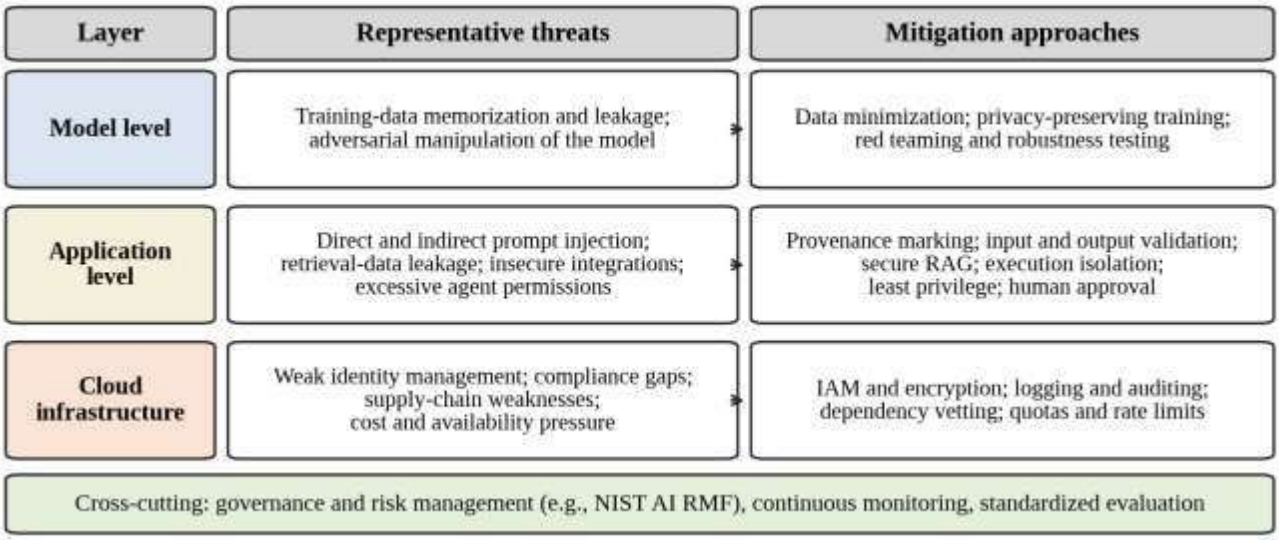


Fig. 2. Conceptual threat and mitigation framework for cloud-hosted Generative AI, organized by model, application, and cloud infrastructure layers.

Model level. *Data privacy and training-data leakage.* LLMs can memorize training data; Carlini *et al.* [17] extracted verbatim sequences, some containing personal information, from GPT-2. In a cloud-hosted service the consequence is exposure of personal or proprietary data in outputs. Defenses are reviewed in [11], [12]; data minimization limits what can leak in the first place (reasoned), at the cost of utility. *Adversarial manipulation.* Jailbreaking and adversarial inputs are surveyed in [12], [13], and Perez and Ribeiro [16] demonstrate goal hijacking and prompt leaking with simple inputs. Red teaming and robustness testing help but cannot prove the absence of vulnerabilities.

Application level. *Prompt injection.* Direct injection places malicious instructions in user input [16]; indirect injection hides them in retrieved web pages, documents, or emails [9]. In a cloud application the injected text can reach the model through any connector, with consequences from altered answers to unauthorized tool calls. OWASP ranks prompt injection first and acknowledges that no fool-proof prevention is known [10]. Spotlighting [18], input filtering, and output checks reduce but do not eliminate the risk. *Retrieval-data leakage and sensitive disclosure.* RAG can expose its private retrieval database [14]; access-controlled retrieval and minimizing stored sensitive data are mitigations, again at some cost to utility (reasoned). *Insecure APIs, integrations, and supply chain.* Third-party apps and plugins may be untrusted; IsolateGPT [19] demonstrates execution isolation, and OWASP lists improper output handling and supply-chain weaknesses [10]. *Excessive agent permissions.* OWASP's Excessive Agency entry covers excessive functionality, permissions, or autonomy [10]. Least privilege and human approval of sensitive actions limit damage but add friction, and reliable authorization models for natural-language-driven agents appear immature (assessment based on [15], [19]).

Cloud infrastructure level. *Weak identity management.* Model endpoints, vector stores, and agents are reached through credentials; over-broad or long-lived credentials let a manipulated component reach more data. Identity and access management (IAM), multi-factor authentication, and short-lived credentials are standard controls, but they do not constrain what a manipulated model requests within its permissions (reasoned). *Compliance and governance.* Data-residency rules, privacy law, and audit duties constrain where prompts and documents may be processed. The NIST Generative AI Profile [20] and AI RMF [21] give voluntary risk-management structure but are not technical enforcement. *Availability, scalability, and cost.* OWASP lists unbounded consumption as a risk [10]; quotas, rate limits, and budget alerts address it, although the security studies reviewed rarely report the cost of controls, with one exception [19]. *Supply chain.* Dependencies include model providers, libraries, datasets, and plugins [10], [20]; vetting and provenance tracking help but depend on provider transparency.

8. COMPARATIVE ANALYSIS OF EXISTING APPROACHES

Table 2 compares ten control categories discussed in the literature and in practitioner guidance. Citations mark rows that draw on a source; the remaining content reflects technical reasoning. No quantitative ranking is attempted because the reviewed studies use different models, attacks, and metrics, so their numbers are not comparable.

Table -2: Security Controls: Benefits, Limitations, and Research Opportunities

Challenge	Mitigation approach	Expected benefit	Main limitation	Implementation consideration	Remaining research opportunity
Prompt injection	Input validation, prompt filtering, provenance marking [18]	Fewer successful instruction hijacks	Tested on limited attack sets; no complete defense [10]	Apply to every untrusted source, including retrieved data	Evaluation against adaptive attackers
Excessive agent permissions	Least-privilege access control	Smaller damage if an agent is manipulated	Hard to scope tools for open-ended tasks	Per-tool, per-task scopes; short-lived credentials	Authorization models for agents
High-impact agent actions	Human approval for sensitive actions	Harmful actions can be caught before execution	Friction; approval fatigue	Define sensitivity thresholds; audit approvals	Usable approval design
Untrusted integrations	Execution isolation with mediated interaction [19]	Contains a compromised app	Overhead; limits collaboration between apps	Fit with managed cloud runtimes	Isolation patterns for cloud AI services
Retrieval and data leakage	Secure RAG: access-controlled retrieval, data minimization [14]	Limits what can be disclosed	Possible loss of utility	Enforce user permissions at retrieval time	Privacy–utility evaluation of RAG defenses
Sensitive or unsafe output	Output validation and filtering [10]	Blocks some leaks and unsafe actions	False positives and negatives	Validate before output reaches tools or users	Reliable leak detection
Weak identity; data exposure	IAM, multi-factor authentication, encryption	Authenticated access; protected stored and transmitted data	Does not constrain a manipulated model within its permissions	Central IAM for models, stores, and agents; key management	Identities for AI agents; protection during inference
Limited visibility	Logging, monitoring, auditing	Detection, forensics, accountability	Logs may hold sensitive prompts; storage cost	Redact and retain by policy	Privacy-preserving telemetry
Unknown weaknesses	Security testing and red teaming [15]	Finds weaknesses before attackers do	Limited coverage; benchmarks may not mirror deployments	Test the whole application, not only the model	Standardized cross-environment benchmarks
Compliance and governance	Risk-management frameworks [20], [21]	Structured, auditable risk management	Voluntary and high-level	Map actions to cloud controls and audits	Evidence of effect on real deployments

Three observations follow. First, controls act at different points. Prompt-level measures such as filtering and provenance marking are probabilistic and reduce the likelihood of manipulation [18], whereas IAM, isolation, and approval workflows are deterministic and limit consequences when manipulation succeeds [19]. Second, none of the reviewed sources claims that a single control eliminates a risk; OWASP notes that prompt injection lacks a fool-proof defense [10]. Third, evidence strength varies: among the reviewed studies, only provenance marking [18] and execution isolation [19] report experimental results, and the other rows rest on guidance [10], [20] and reasoning. Trade-offs include user friction (approvals), latency and cost (isolation and monitoring), and utility loss (filtering and minimized retrieval).

9. FUTURE RESEARCH DIRECTIONS

The directions below are proposals derived from the gaps in Section 4, not established findings.

Robust, evaluated defenses against prompt injection. The problem remains open [9], [10]. Defenses such as provenance marking [18] should be tested against adaptive attackers and across models, with utility cost reported.

Secure RAG architectures. Because retrieval stores can leak [14], work is needed on retrieval-time authorization, per-tenant isolation, and measurable privacy–utility trade-offs.

Least-privilege authorization for AI agents. Isolation [19] and benchmarks [15] are starting points; permission models for natural-language tool use and their integration with cloud IAM are still needed.

Privacy-preserving Generative AI. Training-data extraction [17] motivates privacy-preserving training and inference; the performance and cost impact of such measures should be quantified.

Continuous monitoring and security evaluation frameworks. Standardized, cloud-aware benchmarks should test whole applications and report security, latency, and cost together, while monitoring should detect injection and anomalous agent behavior without retaining sensitive prompts.

Cost- and energy-efficient deployment. Cost and sustainability are noted as challenges [4], and unbounded consumption is a recognized risk [10]; research could link efficiency to security, for example through quotas as a defensive control.

Explainability, trustworthiness, and governance. Governance frameworks [21], [20] need mapping to concrete cloud controls, with evidence that the controls work.

10. DISCUSSION

RQ1. The best-evidenced applications in the reviewed sources are cloud incident analysis [6], [7], code assistance [8], and retrieval-based knowledge access [5]. Agents are emerging [15], and resource management and customer support lack direct evidence here.

RQ2. Prompt injection, sensitive-data disclosure, retrieval-data leakage, and excessive permissions recur across studies [9], [14], [10]. Their shared root is that instructions and data use one channel, so application-level risks persist even when infrastructure is well secured. Cloud choices then amplify or limit the consequences.

RQ3. Mitigations either reduce the likelihood of manipulation (filtering, provenance marking) or limit its consequences (least privilege, isolation, approval, IAM). Experimental evidence exists for few of them [18], [19], and none is complete. Layered use is therefore the defensible position.

RQ4. Section 9 lists directions; the most pressing are evaluated injection defenses, agent authorization, and cross-layer evaluation of security, performance, and cost.

Deployment decisions shape exposure. Whether a model is consumed as a managed service or self-hosted, which data the retrieval layer can reach, which credentials an agent holds, and where logs are stored all change the attack surface. Because of the shared-responsibility boundary, customers should not assume that provider controls cover application-level risks (reasoned). For practitioners, the evidence supports treating all retrieved content as untrusted, scoping agent permissions narrowly, requiring approval for sensitive actions, and

budgeting for consumption. For researchers, it supports evaluations that span layers rather than isolated models.

11. CONCLUSION

This paper reviewed the literature on Generative AI in cloud computing. The reviewed evidence shows demonstrated uses in cloud incident analysis, code assistance, and retrieval-based access to knowledge, with agent-based uses still emerging (RQ1). Recurring risks include prompt injection, sensitive-data disclosure, retrieval-data leakage, and excessive permissions (RQ2). Mitigations reduce either the likelihood or the consequences of attacks, but evaluations are limited and no single control is sufficient (RQ3). The main gaps concern cloud-specific security evaluation, agent authorization, and joint assessment of security, performance, and cost, which define the research directions in Section 9 (RQ4). For cloud service providers, the findings point to secure defaults for retrieval, permissions, and monitoring; for users and organizations, to layered controls and clear data governance.

The study has limitations: it is a preliminary narrative review based on web searches rather than a systematic database search, it covers 21 sources, many results depend on specific models and versions that change quickly, and no experiments were conducted. Future work should extend the search systematically and test the proposed directions empirically.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST Special Publication 800-145, Sep. 2011, doi: 10.6028/NIST.SP.800-145.

[2] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[3] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[4] D. Patel *et al.*, "Cloud platforms for developing generative AI solutions: A scoping review of tools and services," 2024, arXiv:2412.06044 (preprint).

[5] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.

[6] T. Ahmed, S. Ghosh, C. Bansal, T. Zimmermann, X. Zhang, and S. Rajmohan, "Recommending root-cause and mitigation steps for cloud incidents using large language models," in *Proc. IEEE/ACM Int. Conf. Softw. Eng. (ICSE)*, Melbourne, Australia, May 2023, arXiv:2301.03797.

[7] Y. Chen *et al.*, "Automatic root cause analysis via large language models for cloud incidents," in *Proc. Eur. Conf. Comput. Syst. (EuroSys '24)*, Athens, Greece, Apr. 2024, doi: 10.1145/3627703.3629553.

[8] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri, "Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions," *Commun. ACM*, vol. 68, no. 2, pp. 96–105, Feb. 2025.

[9] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proc. ACM Workshop Artif. Intell. Secur. (AISec)*, Nov. 2023, arXiv:2302.12173.

[10] OWASP Foundation, "OWASP Top 10 for LLM applications 2025," Version 2025, Nov. 2024. [Online]. Available: https://genai.owasp.org/

[11] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly," *High-Confidence Comput.*, vol. 4, no. 2, Art. no. 100211, 2024, doi: 10.1016/j.hcc.2024.100211.

[12] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," 2024, arXiv:2402.00888 (preprint).

[13] M. Q. Li and B. C. M. Fung, "Security concerns for large language models: A survey," 2025, arXiv:2505.18889 (preprint).

[14] S. Zeng *et al.*, "The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG)," in *Findings Assoc. Comput. Linguist.: ACL 2024*, Bangkok, Thailand, 2024, pp. 4505–4524, doi: 10.18653/v1/2024.findings-acl.267.

[15] E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Datasets Benchmarks Track*, vol. 37, 2024, arXiv:2406.13352.

[16] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," presented at the NeurIPS ML Safety Workshop, 2022, arXiv:2211.09527.

[17] N. Carlini *et al.*, "Extracting training data from large language models," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 2633–2650.

[18] K. Hines, G. Lopez, M. Hall, F. Zarfati, Y. Zunger, and E. Kiciman, "Defending against indirect prompt injection attacks with spotlighting," 2024, arXiv:2403.14720 (preprint).

[19] Y. Wu, F. Roesner, T. Kohno, N. Zhang, and U. Iqbal, "IsolateGPT: An execution isolation architecture for LLM-based agentic systems," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2025, doi: 10.14722/ndss.2025.241131.

[20] Nat. Inst. Standards Technol., "Artificial intelligence risk management framework: Generative artificial intelligence profile," Gaithersburg, MD, USA, NIST AI 600-1, Jul. 2024, doi: 10.6028/NIST.AI.600-1.

[21] Nat. Inst. Standards Technol., "Artificial intelligence risk management framework (AI RMF 1.0)," Gaithersburg, MD, USA, NIST AI 100-1, Jan. 2023, doi: 10.6028/NIST.AI.100-1.