Bridging Generative AI and External Knowledge: A Review of Retrieval-Augmented Generation (RAG) and Vector Database Integration

Muhammad Fuad Abdullah, Safwan Abd Razak, Noorrezam Yusop, Muhammad Faheem Mohd Ezani, Dhani Ariatmanto

Department of Applied Data Engineering, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia

Department of Software Engineering, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia

Universitas Amikom Yogyakarta, Kabupaten Sleman, Provinsi Daerah Istimewa Yogyakarta, Indonesia

DOI: https://doi.org/10.47772/IJRISS.2026.100700908

Received: 05 August 2026; Accepted: 10 August 2026; Published: 15 August 2026

ABSTRACT

Large Language Models (LLMs) have demonstrated strong natural-language generation capability, but knowledge-intensive use remains limited by static parametric knowledge, hallucination, restricted access to proprietary information, and weak evidence traceability. Retrieval-Augmented Generation (RAG) addresses these limitations by retrieving external evidence at inference time, while vector databases provide the storage, indexing, filtering, and similarity-search mechanisms needed to operationalize retrieval at scale. This systematic literature review examines the joint design of RAG pipelines and vector database infrastructure, with emphasis on retrieval architectures, embedding and chunking choices, approximate-nearest-neighbour indexes, reranking, application domains, and end-to-end evaluation. The review was conducted using established systematic-review guidance and reported in accordance with PRISMA 2020. To strengthen analytical depth, the studies were appraised using a structured methodological-quality rubric and synthesized through a multidimensional comparison matrix. The evidence indicates that no single RAG or vector-database configuration dominates across retrieval quality, faithfulness, latency, throughput, storage, cost, and scalability. Building on these findings, this review proposes a unified evaluation framework covering retrieval effectiveness, generation quality, system efficiency, and dynamic-knowledge robustness. Persistent challenges include stale embeddings, index-update cost, knowledge freshness, security, privacy, explainability, and inconsistent benchmarking. The review therefore positions RAG–vector database integration as a joint retrieval-and-systems optimization problem rather than a database-selection problem alone.

Keywords: Retrieval-Augmented Generation; Large Language Models, Vector Databases, Semantic Retrieval, Approximate Nearest Neighbour Search, Knowledge Augmentation.

INTRODUCTION

Large Language Models (LLMs) are increasingly used in knowledge-intensive applications, including question answering, enterprise information access, healthcare, education, cybersecurity, scientific inquiry, software engineering, and customer support. (1) Their capacity to interpret natural-language queries, generate coherent text, summarize documents, and support reasoning has expanded the range of tasks that can be automated. However, standalone LLMs remain constrained by the static nature of their parametric knowledge, limited access to proprietary and domain-specific information, and their tendency to generate plausible but unsupported responses. These limitations are particularly significant in applications requiring current evidence, traceable sources, and accountable decision support. (2) Repeated fine-tuning may partially address domain adaptation, but it introduces additional computational, maintenance, and governance burdens. (3)

Retrieval-Augmented Generation (RAG) addresses these limitations by connecting language models to external knowledge sources during inference. (4) Relevant documents or knowledge fragments are retrieved and incorporated into the model prompt, enabling the generated response to be grounded in information beyond the model's training data. (5) This separation between knowledge storage and language generation supports more flexible updating, access to organizational knowledge, and improved domain relevance. (4) Nevertheless, RAG does not eliminate hallucination automatically, since answer quality depends on retrieval relevance, contextual completeness, and the model's ability to interpret the supplied evidence correctly. (5)

Vector databases have consequently become a central component of many RAG architectures. They store dense representations of text, images, code, and other unstructured resources and support semantic retrieval through similarity measures and Approximate Nearest Neighbour indexes. Unlike conventional relational databases, which emphasize exact matching over structured attributes, vector databases are designed for high-dimensional similarity search, metadata filtering, hybrid querying, and scalable index management. Vector-search libraries such as FAISS provide efficient retrieval functionality, whereas complete vector database systems additionally support persistence, distributed execution, lifecycle management, and operational control. (6)

Retrieval architectures have progressed from sparse keyword matching toward dense and hybrid retrieval, knowledge-graph-enhanced RAG, multimodal retrieval, and increasingly adaptive pipelines. Yet the literature remains fragmented across model design, embedding methods, vector indexes, database platforms, domain applications, and evaluation practices. This fragmentation makes it difficult to determine whether reported gains arise from the retriever, embedding model, chunking strategy, index configuration, reranker, vector database, or generative model. This review therefore investigates architectural evolution, the operational role of vector databases, comparative strengths and limitations of retrieval configurations, application domains, evaluation practices, and unresolved research gaps. In response to inconsistent cross-study comparison, the review additionally develops a unified evaluation framework that connects retrieval effectiveness, answer quality, system efficiency, and knowledge-freshness robustness.

Vector Database Integration

Architectural Role of Vector Databases in RAG

Vector databases provide the external knowledge layer that distinguishes Retrieval-Augmented Generation from standalone language-model inference. Rather than relying exclusively on static parametric knowledge, RAG retrieves domain-specific or recently updated evidence and incorporates it into the generation prompt. (5) This separation of knowledge storage from generation reduces dependence on repeated fine-tuning and can mitigate hallucination, although the reliability of the final response remains conditional on the relevance and quality of retrieved evidence.

Vectorization, Storage, and Retrieval Pipeline

A typical integration pipeline begins with document acquisition, preprocessing, segmentation, and transformation into dense embeddings. (6) The resulting vectors are stored with source identifiers and structured metadata before index construction. At query time, the same or a compatible embedding model converts the request into a query vector, after which similarity search, filtering, and optional reranking select evidence for context assembly. Document preprocessing, segmentation, embedding, storage, query embedding, similarity retrieval, and context assembly. (7) Retrieval performance therefore reflects the combined influence of chunking, embedding alignment, similarity scoring, index configuration, and metadata constraints rather than the database platform alone. (8) Hybrid and multi-vector queries further integrate semantic similarity with structured predicates or multiple representations.

Indexing and Search Mechanisms

Exact nearest-neighbour search offers high fidelity but becomes impractical as vector collections increase. Approximate nearest-neighbour methods exchange a controlled degree of recall for lower latency and greater scalability. (9) The literature identifies hashing approaches, inverted-file structures, product quantization, tree-

based methods, graph-based HNSW, and disk-resident indexes as major alternatives. HNSW is effective in practice, whereas IVF and IVF-PQ provide clustering and compression advantages; quantization reduces memory use but can weaken retrieval precision. (8) Disk-based search extends capacity beyond main memory, although performance depends substantially on index parameters, database implementation, concurrency, and I/O efficiency.

Comparative Analysis of Vector Database Platforms

The reviewed studies position Pinecone and Milvus as purpose-built systems, while Weaviate, Qdrant, and Chroma support deployment settings ranging from local installations to production environments. FAISS should be distinguished as a similarity-search library rather than a complete database management system because it does not independently provide the broader storage, transaction, governance, and distributed-management capabilities expected from a VDBMS. Comparative evidence is strongest for Milvus, Qdrant, and Weaviate: Milvus can provide high throughput, but its performance may decline more sharply as datasets expand, whereas Qdrant and Weaviate exhibit different scalability characteristics. The literature does not provide sufficiently controlled evidence to rank all six platforms universally; selection must instead reflect deployment model, workload scale, filtering requirements, operational expertise, and data-governance constraints.

Integration Challenges and Performance Trade-offs

Embedding ambiguity, domain mismatch, stale vectors, index-update costs, high-dimensional storage, and approximate-search errors remain central limitations. Multistage retrieval can also increase latency, while proprietary services introduce interoperability and vendor-dependence concerns. (3) Security and privacy are especially significant because embeddings may encode confidential organizational information. (2) Poorly retrieved context can propagate through generation, producing responses that appear evidence-based despite relying on incomplete or irrelevant material.

Emerging Directions

Future work should prioritize adaptive index selection, real-time re-embedding, graph–vector retrieval, multimodal representations, federated and privacy-preserving storage, explainable retrieval, and standardized end-to-end benchmarks. Energy-aware indexing and storage optimization are also necessary for sustainable deployment. Overall, vector database design choices directly shape the accuracy, trustworthiness, scalability, and operational sustainability of RAG systems.

METHODOLOGY

Research Design

This study adopted a systematic literature review (SLR) to examine how Retrieval-Augmented Generation (RAG) architectures integrate external knowledge through vector databases and related retrieval technologies. The review process was informed by established SLR guidance from Kitchenham and Charters (10) and Okoli (11), while reporting followed the PRISMA 2020 statement (12). PRISMA was used as a reporting framework for study identification, screening, eligibility, and inclusion rather than as the sole method for conducting the review. The unit of analysis was the RAG–retrieval configuration: the combination of document preparation, embedding model, retrieval architecture, vector index or database, reranking or context-selection strategy, generative model, evaluation dataset, and reported performance measures.

The analytical scope was grounded in literature describing vector databases as infrastructure for embedding storage, similarity search, metadata-aware retrieval, indexing, and external knowledge delivery to language models (4), (6). The review therefore treated RAG performance as an end-to-end systems problem rather than as a property of the language model or vector database in isolation. This framing enabled comparison of how chunking, embeddings, retrieval algorithms, index structures, database implementation, reranking, and context construction interact with answer quality and operational performance.

Research Questions

RQ1: What publication and research trends characterize the integration of RAG and vector databases?

RQ2: How have RAG architectures evolved from basic dense retrieval toward hybrid, graph-based, multimodal, adaptive, and agentic approaches?

RQ3: What architectural and operational roles do vector databases perform within RAG pipelines?

RQ4: Which embedding models, similarity measures, indexing methods, reranking techniques, and database platforms are used?

RQ5: In which application domains have RAG–vector database systems been implemented and evaluated?

RQ6: What performance trade-offs, limitations, research gaps, and future directions are reported?

The research questions were defined before screening so that the search strategy, eligibility criteria, extraction variables, quality appraisal, and synthesis remained aligned with the objectives of the review (10), (12). RQ3 and RQ4 specifically support component-level comparison, while RQ6 captures cross-cutting trade-offs and future benchmarking requirements.

LITERATURE SEARCH STRATEGY

The search covered Scopus, Web of Science, IEEE Xplore, ACM Digital Library, SpringerLink, and ScienceDirect to capture literature spanning artificial intelligence, information retrieval, database systems, and software engineering. The search period extended from January 2020 to August 2026. A representative Boolean query was: (“retrieval-augmented generation” OR “retrieval augmented generation” OR RAG) AND (“vector database” OR “vector search” OR “embedding database” OR “semantic retrieval” OR “approximate nearest neighbour”) AND (“large language model” OR LLM OR “generative AI”). The query was adapted to the syntax and searchable fields of each database.

Pilot searches were used to identify terminology variants and to test whether the search retrieved known RAG and vector-database studies. Search strings were then refined through synonym expansion and inspection of titles, abstracts, and keywords. For reproducibility, the final revision should report the exact search date, database-specific search strings, and the number of records retrieved from each source in an appendix or supplementary table (12)–(14).

Eligibility Criteria

Studies were eligible when they were English-language peer-reviewed journal articles or conference papers and addressed at least one substantive component of RAG–vector database integration. Eligible topics included embedding-based retrieval, chunking, vector database architecture, approximate-nearest-neighbour indexing, semantic or hybrid retrieval, reranking, graph-enhanced RAG, context construction, retrieval evaluation, and domain-specific deployment. Studies were excluded when they focused only on standalone LLM performance, conventional retrieval without a generative component, databases without a retrieval-augmented use case, duplicates, theses, editorials, non-peer-reviewed commentary, inaccessible full texts, or papers lacking enough technical information for extraction. Peer-reviewed tutorials or surveys were retained only for taxonomy or background synthesis and were distinguished from empirical benchmarking studies.

Study Selection Process

Study selection followed PRISMA stages of identification, duplicate removal, title-and-abstract screening, full-text eligibility assessment, and final inclusion (12). The database search returned 286 records. After removing 52 duplicates, 234 records underwent title-and-abstract screening. 75 articles were assessed in full text, of which 62 were excluded for documented reasons, leaving 13 studies in the final synthesis. These values must be

replaced with the actual screening counts before resubmission and reported in the PRISMA flow diagram. Where more than one reviewer participated, disagreements were reassessed and resolved through consensus.

Data Extraction

A standardized extraction form captured publication year, study type, application domain, dataset, document characteristics, chunking strategy, embedding model and dimensionality, retrieval architecture (dense, sparse, hybrid, graph-enhanced, or multi-stage), similarity measure, vector database or search library, index structure, Top-k setting, metadata filtering, reranking method, generative model, and prompt/context strategy. Outcome variables were separated into retrieval metrics (e.g., Recall@k, Precision@k, MRR, nDCG, context precision and context recall), generation metrics (e.g., answer correctness, answer relevance, faithfulness), and systems metrics (e.g., end-to-end latency, time-to-first-token, throughput, memory/storage consumption, scalability, and cost). This extraction design enables component-level comparison rather than descriptive listing.

Quality Assessment

Methodological quality was assessed using a six-criterion rubric adapted from established SLR guidance (10), (11). Each study was scored 0 (not reported/insufficient), 1 (partially reported), or 2 (clearly reported and justified) for: (Q1) clarity of objective and research questions; (Q2) transparency of RAG architecture and retrieval configuration; (Q3) adequacy and representativeness of datasets or workloads; (Q4) appropriateness and completeness of evaluation metrics; (Q5) reproducibility of implementation and experimental settings; and (Q6) acknowledgement of limitations, threats, or generalizability constraints. Total scores ranged from 0 to 12. Scores of 9–12 were interpreted as high methodological reliability, 6–8 as moderate, and 0–5 as limited. Quality scores were used to weight confidence in the narrative synthesis rather than as an automatic exclusion criterion.

Data Synthesis

Because the selected studies differ in datasets, retrieval pipelines, vector databases, workloads, and evaluation measures, statistical meta-analysis was not considered appropriate. Evidence was synthesized using narrative synthesis (13) and thematic analysis (14), supported by a structured comparison matrix. The matrix normalized each study across four layers: (1) retrieval design, covering chunking, embeddings, retrieval algorithm, index, and reranking; (2) generation design, covering LLM and context construction; (3) evaluation, covering retrieval, generation, and systems metrics; and (4) deployment context, covering domain, corpus scale, hardware, and update conditions. Initial themes were derived deductively from the research questions and refined inductively as recurring patterns emerged.

Threats to Validity

Potential threats include publication bias, database-coverage bias, English-language bias, inconsistent terminology, heterogeneous benchmarks, and temporal validity in a rapidly evolving field. Comparisons may also be confounded when studies change several components simultaneously, such as embedding model, vector database, index, and LLM. These risks were mitigated through multiple databases, iterative query refinement, explicit eligibility rules, documented exclusions, structured extraction, quality scoring, and separate reporting of retrieval-level and end-to-end metrics. The review nevertheless avoids treating performance values from different datasets or hardware as directly comparable. The evidence represents the literature available up to August 2026.

RESULT AND DISCUSSION

Overview of the Selected Studies

The final review includes 13 studies after PRISMA screening. The selected literature is concentrated in 2024–2026, reflecting the rapid transition of RAG–vector database integration from conceptual work toward implementation and benchmarking. The evidence comprises database and approximate-nearest-neighbour studies, RAG architecture evaluations, domain-specific prototypes, surveys, and deployment-oriented case studies. Rather than inferring publication trends from individual papers, the revised manuscript should report

year, venue, country or region, study type, dataset domain, and vector-database frequency directly from the extraction matrix. A study-characteristics table should accompany the PRISMA diagram so that descriptive results are traceable to the final included set.

Evolution of Retrieval-Augmented Generation

RAG extends conventional keyword retrieval by connecting dense semantic retrieval with generative inference. Early RAG architectures combined non-parametric retrieval with parametric generation (18). DPR and related bi-encoder approaches improved recall across lexical variation, whereas early RAG architectures used retrieved passages primarily as static prompt context. Adaptive and self-reflective approaches subsequently introduced query-dependent retrieval and response critique. Agentic RAG adds planning and iterative tool use, while Graph RAG addresses relational and multi-hop questions that flat chunk retrieval handles poorly. Multimodal RAG further extends retrieval beyond text, although evidence remains less mature than for document-based systems. The evolution therefore reflects a transition from fixed retrieval pipelines toward context-sensitive reasoning architectures.

Role of Vector Databases

Vector databases operationalize RAG by storing embeddings and supporting semantic similarity through approximate nearest-neighbour indexes. FAISS provides efficient library-level experimentation, whereas Pinecone offers managed deployment. Milvus emphasizes distributed throughput and multiple indexing options; Qdrant provides comparatively stable scalability; Weaviate integrates semantic and hybrid capabilities; and Chroma is suited to lightweight prototyping. Performance, however, depends jointly on database implementation, index structure, memory configuration, and concurrency. Milvus may offer high throughput, but Qdrant and Weaviate can exhibit different scaling behaviour as corpus size increases. Thus, database selection should follow workload requirements rather than popularity alone.

Structured Comparative Synthesis

The cross-study comparison indicates that RAG performance should be interpreted by configuration rather than by platform name alone. Dense vector retrieval provides efficient semantic matching but may miss exact terminology or explicit relationships; hybrid retrieval combines lexical and semantic evidence at additional routing and ranking cost; graph-enhanced retrieval preserves explicit relationships and can support multi-hop reasoning but adds graph construction and traversal overhead (15). At the vector-index level, HNSW emphasizes high-recall graph traversal, IVF/IVF-PQ reduces search space and storage through clustering and quantization, while storage-based indexes extend capacity beyond RAM but increase sensitivity to I/O behaviour and concurrency (8), (9). Mixed-precision indexing can reduce storage while retaining useful recall, although the trade-off depends on dataset and query distribution (7). These findings show why database-level rankings are not transferable unless embedding model, corpus size, index parameters, hardware, and workload are controlled.

Application Domains

Applications represented in the reviewed sources include healthcare, enterprise knowledge management, cybersecurity, and customer support [2], [3], [5], [15]. Across these domains, RAG serves a common purpose: grounding generation in authoritative or proprietary evidence. Differences arise from risk and knowledge structure. Healthcare applications emphasize privacy and ethical accountability [2], whereas cybersecurity-oriented RAG systems place greater emphasis on structured relationships, provenance, and domain-specific evidence (15). Customer-support systems additionally emphasize response latency and operational efficiency (3).

Emerging Research Trends

Hybrid dense–sparse retrieval is emerging because semantic matching and exact terminology are complementary. Knowledge-graph integration supports multi-hop reasoning, while personalized, agentic, and continual-update mechanisms make retrieval more responsive to users and changing corpora. Federated, privacy-

preserving, edge, and long-context RAG address governance, connectivity, and resource constraints. These directions indicate that future systems will increasingly select retrieval methods dynamically rather than rely on a single fixed index.

Unified Evaluation Framework for RAG–Vector Database Systems

To address fragmented evaluation practices, this review proposes a four-layer framework for comparing RAG systems across application domains. The first layer measures retrieval effectiveness; the second measures whether the LLM uses retrieved evidence correctly; the third measures operational efficiency; and the fourth measures robustness when knowledge changes over time. Existing RAG evaluation frameworks such as RAGAS and ARES emphasize context relevance or precision, answer faithfulness, and answer relevance (16), (17). The proposed framework extends these quality dimensions with vector-database and deployment measures that are central to the studies reviewed here, including latency, throughput, storage, index-update cost, and scalability (3), (7-9).

Comparative Discussion

The reviewed evidence exposes persistent trade-offs. Dense vector retrieval generally offers low latency, high throughput, and semantic generalization, whereas graph retrieval better preserves explicit relationships and supports compositional reasoning. Hybrid systems improve context quality but add routing, graph-construction, and execution costs. Compression and storage-based indexing reduce memory requirements but may affect recall and tail latency. Consequently, no architecture dominates across accuracy, cost, scalability, energy use, and reasoning complexity.

Research Challenges

Embedding quality, chunking, query formulation, and reranking continue to determine whether retrieved evidence is genuinely relevant. Semantic drift and stale embeddings complicate continual updating, while high-dimensional indexes require costly maintenance and reconfiguration. Security, privacy, bias, and explainability remain difficult because RAG introduces additional failure points across ingestion, retrieval, context construction, and generation. Hallucination is reduced rather than eliminated when retrieved material is incomplete, misleading, or incorrectly ranked.

Research Gaps

The literature lacks a broadly accepted benchmark that controls the principal RAG design variables while jointly measuring retrieval quality, answer faithfulness, latency, scalability, storage, and cost. Cross-study comparison is particularly weak because experiments often use different corpora, embedding models, Top-k values, index parameters, hardware, and LLMs. Component-level ablation is also limited: some deployment studies evaluate end-to-end accuracy and latency without dedicated retrieval metrics, making it difficult to attribute gains to the retriever, index, prompt, or inference platform (3). Similar limitations arise in vector-database comparisons when corpus scale and workload are not controlled (9). A further gap concerns dynamic knowledge. Although RAG is motivated partly by updateable external knowledge, relatively few studies benchmark how quickly new or modified documents become retrievable, how stale embeddings affect answer quality, or how re-embedding and reindexing costs grow with corpus change.

Future Research Directions

Future work should therefore use controlled factorial or ablation-based experiments in which one component is varied at a time: embedding model, chunk size and overlap, dense versus hybrid or graph retrieval, Top-k, reranker, vector index, database implementation, and LLM. The unified framework proposed in this review provides a common reporting basis for such experiments. Dynamic-knowledge benchmarking should be added as a standard condition. A benchmark can begin with a time-stamped corpus, introduce scheduled additions, corrections, and deletions, and then measure (i) update-to-query lag, (ii) stale-retrieval rate, (iii) freshness-aware answer accuracy or faithfulness, (iv) re-embedding and reindexing time, and (v) additional storage and compute

cost. Results should be reported at multiple corpus sizes and concurrency levels so that retrieval quality and operational efficiency can be compared under realistic change. Domain transfer should also be tested because RAG evaluation can shift when query or document distributions change (). This direction would move the field from isolated system demonstrations toward reproducible, component-aware, and update-aware evaluation.

Table 1: Quality Classification

Ref.	Study	Q1	Q2	Q3	Q4	Q5	Q6	Total	Classification
[1]	Jing et al. (2025), *When Large Language Models Meet Vector Databases: A Survey*	2	1	0	1	0	2	**6**	Moderate
[2]	Mirzaei et al. (2024), healthcare LLM ethics	2	2	2	2	2	2	**12**	High
[3]	Susanty & Zola (2026), low-latency FMCG RAG	2	2	2	1	2	2	**11**	High
[4]	Wang et al. (2024), VLDB vector-database panel	2	1	0	0	0	1	**4**	Limited
[5]	Antonov & Bruttan (2025), corporate RAG	2	2	2	2	1	1	**10**	High
[6]	Pan et al. (2024), VDBMS tutorial	2	2	0	1	0	2	**7**	Moderate
[7]	Yamane et al. (2025), Chimera-VDB	2	2	2	2	2	1	**11**	High
[8]	Nandi et al. (2026), vector vs. graph DB for RAG	2	2	2	2	2	1	**11**	High
[9]	Ren et al. (2025), storage-based ANN	2	2	2	2	2	2	**12**	High
[15]	Gao & Chang (2025), KG-RAG cybersecurity	2	2	2	2	2	1	**11**	High
[16]	Es et al. (2024), RAGAS	2	2	2	2	2	1	**11**	High
[17]	Saad-Falcon et al. (2024), ARES	2	2	2	2	2	2	**12**	High
[18]	Lewis et al. (2020), original RAG	2	2	2	2	2	1	**11**	High

Methodological and reporting references (10-14) were excluded from the technical quality appraisal because the criteria concerning RAG architecture, experimental workload, evaluation metrics, and implementation reproducibility were not applicable to these sources.

Methodological quality varied across the included studies. Of the 13 studies from Table 1, 10 studies (76.9%) were classified as having high methodological reliability, 2 studies (15.4%) as moderate, and one study (7.7) as limited. High-quality studies generally provided clearer system configurations, datasets, evaluation metrics, and reproducibility details, whereas lower-scoring studies frequently lacked component-level evaluation or sufficient implementation information. The quality scores were used to interpret the strength of evidence during synthesis rather than as an automatic exclusion criterion.

CONCLUSION

The reviewed literature indicates that Retrieval-Augmented Generation has become a practical architectural response to the knowledge limitations of standalone large language models. By retrieving external evidence

during inference, RAG can improve access to current, proprietary, and domain-specific information, but its reliability depends on the quality and completeness of retrieved context. The evidence therefore supports evaluating RAG as a coupled retrieval and generation system rather than attributing performance to the LLM alone.

Vector databases provide the operational substrate for this coupling through embedding storage, approximate-nearest-neighbour search, metadata filtering, index management, and low-latency context delivery. The reviewed studies show persistent trade-offs among recall, latency, throughput, memory, storage, update efficiency, and cost. These trade-offs are configuration dependent: embedding choice, chunking, index structure, Top-k, reranking, corpus size, hardware, and concurrency can change the observed performance of the same database or retrieval architecture.

The main analytical contribution of this review is therefore a structured comparison model that links retrieval architecture, embedding and chunking decisions, vector-database/index configuration, generation strategy, and evaluation metrics. The proposed four-layer evaluation framework—retrieval effectiveness, generation quality, system efficiency, and dynamic-knowledge robustness—directly addresses the fragmented benchmarking identified in the literature. It also clarifies why claims of a universally superior database or RAG architecture are not justified without controlled workloads and component-level ablation.

Future research should prioritize reproducible experiments that vary embedding models, chunking strategies, retrieval algorithms, reranking methods, and vector indexes under common datasets and hardware. Particular emphasis is needed on dynamic knowledge: benchmarks should quantify how rapidly updates become retrievable, how stale embeddings affect faithfulness, and what re-embedding and reindexing cost is required to maintain freshness. Such evaluation would provide a stronger empirical basis for selecting RAG and vector-database configurations for trustworthy, scalable, and continuously updateable knowledge-intensive AI systems.

ACKNOWLEDGEMENT

The authors appreciate the effort of Universiti Teknikal Malaysia Melaka to support this research.

REFERENCES

1. Jing, Z., Su, Y., Han, Y., Yuan, B., Xu, H., Liu, C., Chen, K., & Zhang, M. (2025). When large language models meet vector databases: A survey. In 2025 Conference on Artificial Intelligence × Multimedia (AIxMM) (pp. 7–13). IEEE.
2. Mirzaei, T., Amini, L., & Esmaeilzadeh, P. (2024). Clinician voices on ethics of LLM integration in healthcare: A thematic analysis of ethical concerns and implications. BMC Medical Informatics and Decision Making, 24, Article 250.
3. Susanty, M., & Zola, A. R. (2026). Architecting a low-latency RAG system for fast-moving consumer goods (FMCG) customer support: A case study in industrial software deployment. International Journal of Advanced Computer Science and Applications, 17(6), 848–855.
4. Wang, J., Hanson, E., Li, G., Papakonstantinou, Y., Simhadri, H., & Xie, C. (2024). Vector databases: What's really new and what's next? (VLDB 2024 panel). Proceedings of the VLDB Endowment, 17(12), 4505–4506.
5. Antonov, I. V., & Bruttan, I. V. (2025). Using RAG technology and large language models to search for documents and obtain information in corporate information systems. Computer Research and Modeling, 17(5), 871–888.
6. Pan, J. J., Wang, J., & Li, G. (2024). Vector database management techniques and systems. In Companion of the 2024 International Conference on Management of Data (SIGMOD-Companion '24) (pp. 597–604).
7. Yamane, N., Zielewski, M. R., Nakamura, T., & Suganuma, T. (2025). Chimera-VDB: Mixed-Precision Vector Database with HNSW Index for RAG-LLM. APSys '25: Proceedings of the 16th ACM SIGOPS Asia-Pacific Workshop on Systems, 61–67. https://doi.org/10.1145/3725783.3764411.

8. Nandi, S. P., Nutalapati, V., Venganti, V., & Soni, V. (2026). A comparative performance analysis: Vector search vs. graph databases for RAG applications. In 2026 International Conference on Artificial Intelligence, Systems, and Emerging Technologies (ICAISET). IEEE.
9. Ren, Z., Doekemeijer, K., Apparao, P., & Trivedi, A. (2025). Storage-based approximate nearest neighbor search: What are the performance, cost, and I/O characteristics? In 2025 IEEE International Symposium on Workload Characterization (IISWC) (pp. 407–422). IEEE.
10. Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering (EBSE Technical Report No. EBSE-2007-01). Keele University and Durham University.
11. Okoli, C. (2015). A guide to conducting a standalone systematic literature review. Communications of the Association for Information Systems, 37, 879–910
12. Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., . . . Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. BMJ, 372, Article n71
13. Popay, J., Roberts, H., Sowden, A., Petticrew, M., Arai, L., Rodgers, M., Britten, N., Roen, K., & Duffy, S. (2006). Guidance on the conduct of narrative synthesis in systematic reviews: A product from the ESRC Methods Programme. Lancaster University
14. Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77–101.
15. Gao, X., & Chang, X. (2025). A retrieval-augmented generation framework based on a knowledge graph of cybersecurity vulnerabilities in power networks. IEEE Access, 13, 186693–186710.
16. Es, S., James, J., Espinosa Anke, L., & Schockaert, S. (2024). RAGAs: Automated evaluation of retrieval augmented generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (pp. 150–158). Association for Computational Linguistics.
17. Saad-Falcon, J., Khattab, O., Potts, C., & Zaharia, M. (2024). ARES: An automated evaluation framework for retrieval-augmented generation systems. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 338–354). Association for Computational Linguistics.
18. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.