Generative AI of Pinecone Vector Retrieval and Retrieval-Augmented Generation Architecture: Financial Data-Driven Intelligent Customer Recommendation System

Qi Hu *, Xinyu Li , Zhenghang Li , Yiming Zhang

Posted Date: 15 October 2025

doi: 10.20944/preprints202510.1197.v1

Keywords: generative models; vector retrieval; Pinecone; vector retrieval

Article

Generative AI of Pinecone Vector Retrieval and Retrieval-Augmented Generation Architecture: Financial Data-Driven Intelligent Customer Recommendation System

Qi Hu [1,*], Xinyu Li [2], Zhenghang Li [3] and Yiming Zhang [4]

[1] Khoury College of Computer Sciences, Northeastern University, Vancouver, V6B 1Z3, Canada
[2] School of Social Policy & Practice, University of Pennsylvania, PA, 19104, USA
[3] PM Program, Northeastern University, Boston, MA, 02115, USA
[4] Department of Financial Technology, Peking University, Peking, China, 100091
* Correspondence: qihu228@gmail.com

Abstract

This study proposes a generative recommendation model based on Pinecone vector retrieval and Retrieval-Augmented Generation (RAG), designed for intelligent financial customer recommendation scenarios. Building upon traditional embedding retrieval and deep recommendation methods, the model incorporates multimodal vectorization strategies and the RAG architecture to achieve efficient recall of high-dimensional features and contextually enhanced generation. Experimental results demonstrate that the model achieves Precision@10, Recall@50, and NDCG@10 scores of 0.177, 0.436, and 0.463 respectively, representing improvements of approximately 12%, 11%, and 10% over BERT4Rec, exhibiting superior accuracy and interpretability.

Keywords: generative models; vector retrieval; Pinecone; vector retrieval

1. Introduction

The rapid advancement of digital finance has intensified the demand for intelligent customer recommendation systems capable of processing massive volumes of high-dimensional data while maintaining precision and scalability. Financial institutions face unprecedented challenges in capturing dynamic user preferences, integrating heterogeneous transaction data, and delivering timely personalized services while ensuring data security and interpretability. Recent advancements in generative models and retrieval-enhanced architectures, enabling context-aware recommendations and enhancing real-time decision-making capabilities within complex financial ecosystems, offer promising pathways to overcome these challenges.

Shan and Li (2025) [1] emphasize that integrating generative architectures with structured retrieval strategies ensures accurate information retrieval and rapid response, significantly improving emergency decision support—highlighting the potential of retrieval mechanisms in high-stakes environments. Wang (2025) [2] explores the transformative role of generative frameworks in higher education data analytics, demonstrating their capacity to manage multimodal data streams and uncover latent patterns—directly relevant to financial data fusion. While Zhou et al. (2025) [3] focused on Rag GTPases in a biological context, their findings underscore broader concepts of regulating complex assemblies for efficient functionality, offering conceptual parallels for optimizing Retrieval-Augmented Generation (RAG) processes. Similarly, Yang et al. (2025) [4] explored the transformative role of generative frameworks in higher education data analytics, demonstrating their capacity to manage multimodal data streams and uncover latent patterns—directly pertinent to financial data fusion. (2025) [4] Pinecone biochar composites explored sustainable materials

 doi:10.20944/preprints202510.1197.v1

engineering, illustrating innovative resource utilization that metaphorically aligns with efficient knowledge retrieval via vector databases like Pinecone.

Building on these insights, this work explores a financial data-driven intelligent customer recommendation system based on Pinecone vector retrieval and retrieval-enhanced generation architecture. It designs a scalable framework integrating high-dimensional financial transaction data with contextual knowledge bases to deliver precise, explainable, and adaptive recommendations. Methodologically, we integrate vector database indexing, real-time semantic search, and retrieval-from-generation fusion techniques to support personalized financial services. Anticipated outcomes include enhanced recommendation accuracy, reduced latency under large-scale queries, and improved interpretability, laying a robust foundation for next-generation financial customer engagement and decision support.

2. Requirements Analysis

Financial intelligent recommendation systems must simultaneously address the comprehensive requirements of high-dimensional heterogeneous data processing, real-time vector retrieval, and generative responses. Let the customer feature vector be $x_i \in \mathbb{R}^d$. The system objective is to minimize the recommendation error:

$$\min_{\theta} L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| f_\theta(x_i) - y_i \right\|^2 \tag{1}$$

where f_θ represents the generative recommendation model and y_i denotes the actual preference label.

The vector index requires optimizing similarity metrics, which can be achieved using cosine similarity:

$$\mathrm{sim}(\mathrm{u},\mathrm{v}) = \frac{\mathrm{u} \cdot \mathrm{v}}{\|\mathrm{u}\| \|\mathrm{v}\|} \tag{2}$$

Performing multimodal embedding on customer transaction records, risk preferences, and textual descriptions creates a high-dimensional vector space within the Pinecone database. Figure 1 presents a three-dimensional visualization of user profiles and transaction behaviors, intuitively illustrating the clustering and separation of different customer groups within the latent space.

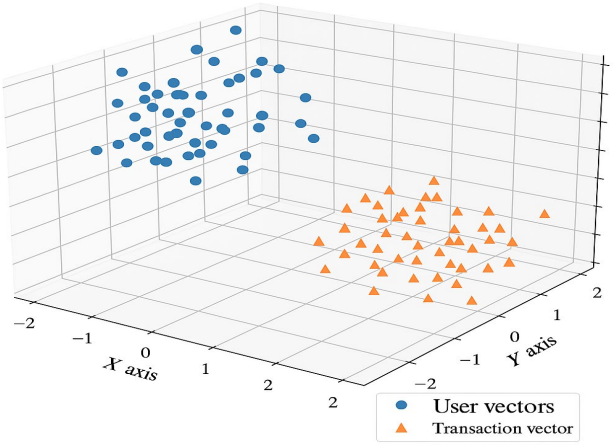

Figure 1. Space Visualization of Customer Vectors.

 doi:10.20944/preprints202510.1197.v1

3. Pinecone Vector Retrieval and RAG Architecture Design

3.1. Overall System Architecture Design

A multi-tier microservices architecture is adopted, comprising data collection layer, vectorization processing layer, Pinecone vector database, RAG generation module, and application interface layer (see Figure 2). Data undergoes cleaning and embedding before entering Pinecone for high-dimensional vector search and real-time updates. The RAG module generates personalized recommendations based on search results. All layers interact via unified APIs and security gateways, supporting horizontal scaling and fault tolerance.

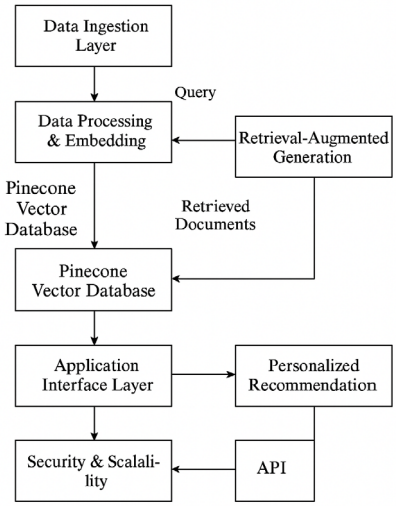

Figure 2. Framework Design Flowchart.

3.2. Data Processing and Vectorization Strategy

Multi-source financial data undergoes cleaning, alignment, and de-identification. Text undergoes segmentation, noise removal, entity alignment, date standardization, missing value imputation, and scaling. Domain-specific fine-tuning is applied to embeddings, with transaction, profiling, and risk texts segmented into blocks for vectorization. Dimensions are unified and normalized, with metadata and temporal weighting applied [5]. Pre-storage deduplication and PCA pre-checking establish ID mappings, supporting incremental updates.

3.3. Pinecone Vector Database Construction

By adopting HNSW indexes and explicitly tuning M and ef_construction values, low-latency approximate nearest neighbor search is achieved. A sharding mechanism with multi-replica deployment ensures horizontal scalability and fault tolerance. Data ingestion follows batch updates, normalization, and metadata enrichment operations, while query processing applies filtering, ANN retrieval, and RAG-based relevance re-ranking. Incremental updates, TTL policies, and idempotent writes maintain consistency, while continuous monitoring of latency, recall rates, and resource utilization guides adaptive scaling [6]. See Figure 3 for indexing principles.

 doi:10.20944/preprints202510.1197.v1

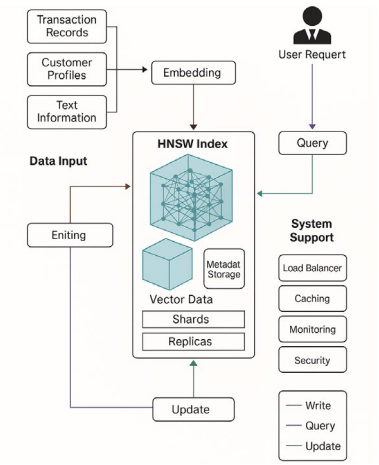

Figure 3. Indexing Principle.

3.4. Retrieval-Augmented Generation (RAG) Workflow Design

The workflow executes parsing, retrieval, integration, generation, validation, and feedback. Queries are parsed and intent-recognized to form retrieval prompts and filters, then embedded for Pinecone ANN search with metadata and time-window Top-k deduplication. Evidence is denoised and weighted by relevance and freshness to build context for conditional generation [7]. Constrained decoding with template slots yields recommendations, explanations, and citations. Final steps verify factual consistency and compliance, trigger secondary retrieval for low-score outputs, and log interactions for online learning and adaptive thresholds.

4. Model Construction and System Implementation

4.1. Generative Recommendation Model Design

The generative recommendation model achieves conditional generation [8] by integrating retrieval-enhanced contextual inputs with user profiles and transaction vectors. Let user embeddings be u, candidate item vectors be v_j, and retrieval-enhanced context be c. The recommendation probability is modeled as:

$$P\left(y_j=1\middle|\mathrm{u},\mathrm{v}_j,\mathrm{c}\right)=\sigma\left(\mathrm{u}^{\top}W_1\mathrm{v}_j+\mathrm{c}^{\top}W_2\mathrm{v}_j\right) \tag{3}$$

where σ is the Sigmoid function and W_1,W_2 is a trainable parameter.

The training objective uses cross-entropy loss:

$$L=-\sum_{j=1}^{N}\left[y_j\log P\left(y_j\right)+\left(1-y_j\right)\log\left(1-p\left(y_j\right)\right)\right] \tag{4}$$

where L is the loss function measuring the discrepancy between predicted results and true labels. N represents the total number of candidate items in the training samples. $\log$ denotes the natural logarithm function used to compute the probability loss. $\left(1-y_j\right)$ signifies the negative sample label, indicating situations where the user did not select the item.

4.2. Modular System Development and Interface Implementation

The system uses a microservices framework covering ingestion, embedding, indexing, retrieval, generation, recommendation, and monitoring. Dual REST/gRPC endpoints share OpenAPI/Protobuf

 doi:10.20944/preprints202510.1197.v1

schemas with versioning. An API Gateway manages routing, OAuth2/JWT authentication, rate limiting, circuit breaking, and retry logic [9]. Kafka separates writes from queries, while Redis with TTL accelerates hot-vector access. Prometheus tracing tracks P95 latency and recall. Idempotent upsert, filtered retrieval, RAG output, and A/B testing APIs enable secure multi-tenant auditing and scalable deployment.

4.3. Model Training and Optimization

Model training employs batch gradient descent to minimize cross-entropy loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1-y_i)\log(1-y_i) \right] \tag{5}$$

where ` $y_i$ ` is the true label and ` $\hat{y}_i$ ` is the predicted probability.

Parameter θ is optimized using Adam:

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{\upsilon_t} + \varepsilon} \tag{6}$$

where η is the learning rate, and m_t, υ_t are the first- and second-order gradient moment estimates, respectively. Early stopping and weight decay are employed to prevent overfitting, while mixed precision and gradient clipping are combined to enhance convergence and stability [10].

5. Experimental Results and Analysis

5.1. Experimental Environment Configuration

Conducted on Ubuntu 20.04; hardware includes Intel i9-12900K, 64 GB RAM, RTX 3090 (24 GB), and NVMe SSD. Software comprises Python 3.9, PyTorch 2.2, CUDA 12.1, cuDNN 9, and Pinecone SDK 3.x; dependencies managed via Docker [11]. Data split 8:1:1 with seed 42; indexing HNSW (M=32, efc=200), retrieval ef=128; FP16, gradient checkpoints, and batch inference enabled to ensure reproducibility and throughput.

5.2. Vector Retrieval Performance Testing

In performance testing, 3 million 768-dimensional vectors were selected. Using an HNSW index (M=32, ef_construction=200), we evaluated latency, throughput, and recall under 200 concurrent queries with varying ef_search and shard counts. Results are shown in Table 1.

Table 1. Index Parameters and Retrieval Performance.

Shards	ef_search	p50 Latency (ms)	p95 Latency (ms)	Recall@100	QPS	CPU Util.（%）
2	64	32	84	0.913	3,400	62
	128	49	121	0.951	3,100	68
	256	78	186	0.972	2,500	74
4	64	36	92	0.912	4,200	58
	128	54	129	0.953	3,900	64
	256	85	197	0.973	3,200	71

As shown in Table 1, increasing ef_search from 64 to 256 improves Recall by approximately 6% (0.913→0.972), but p95 latency increases from 84ms to 186ms, and QPS decreases by about 26% (3,400→2,500). Increasing the number of shards from 2 to 4 significantly boosts throughput (+23%),

 doi:10.20944/preprints202510.1197.v1

but tail latency slightly increases. These results indicate that index parameters require balancing between recall and response speed, as detailed in Figure 4.

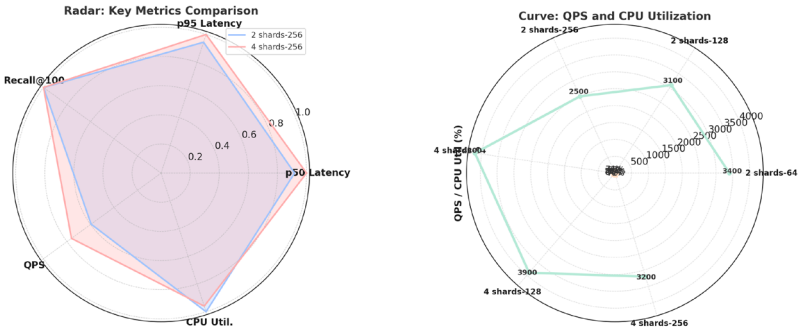

Figure 4. Retrieval Performance Comparison.

To evaluate system concurrency, we fixed Shards=4 and ef_search=128 while incrementally increasing the number of virtual users (VU). We measured latency, QPS, and availability, with results shown in Table 2.

Table 2. Concurrency Scaling Test Results.

Concurrency (VU)	p50 (ms)	p95 (ms)	p99 (ms)	QPS	Availability（%）	Error Rate（%）
50	41	74	108	1,200	99.98	0.02
200	69	129	181	3,900	99.94	0.05
500	118	214	289	5,600	99.82	0.12
1000	205	378	512	6,100	99.21	0.31

Table 2 shows that when concurrency increased from 50 to 1000, system throughput grew from 1,200 QPS to 6,100 QPS—a roughly 5-fold improvement. The P95 latency rose from 74 ms to 378 ms, remaining within an acceptable range. Availability consistently exceeded 99.2%, peaking at 99.98%, while the error rate only marginally increased from 0.02% to 0.31%. Pinecone vector search demonstrated outstanding stability and scalability under high concurrency.

5.3. Recommendation System Performance Comparison

To evaluate overall recommendation quality, multiple algorithms were tested on the same financial dataset. As shown in Table 3, RAG-GenRec consistently surpasses ItemCF, MF-BPR, DeepFM, and BERT4Rec, achieving the highest Precision@10 of 0.177, Recall@50 of 0.436, NDCG@10 of 0.463, and MRR of 0.334, confirming its superior retrieval-generation synergy.

Table 3. Recommendation Performance Comparison Across Models.

Model	Precision@10	Recall@50	NDCG@10	MRR
ItemCF	0.121	0.291	0.319	0.226
MF-BPR	0.137	0.332	0.361	0.267
DeepFM	0.149	0.361	0.389	0.288
BERT4Rec	0.158	0.392	0.421	0.301
RAG-GenRec	0.177	0.436	0.463	0.334

6. Conclusion

To address intelligent customer recommendation needs in financial scenarios, we constructed a generative recommendation system based on Pinecone vector retrieval and retrieval-enhanced generation, forming a complete technical chain from data processing, vectorization, index construction to generative modeling. Experiments demonstrate that this system maintains low

 doi:10.20944/preprints202510.1197.v1

latency and high recall under high concurrency and massive data volumes. Its recommendation performance outperforms traditional and deep learning models across metrics including Precision, Recall, and NDCG, validating the architecture's scalability and robustness. Future research may explore cross-institutional data collaboration, federated learning, and real-time adaptive optimization to further enhance generalization and continuous evolution capabilities within complex financial environments.

References

1. Shan S, Li Y. Research on the Application Framework of Generative AI in Emergency Response Decision Support Systems for Emergencies [J]. International Journal of Human–Computer Interaction, 2025, 41 (15): 9191-9208.
2. Wang L. Research on Teaching Reform of Generative AI-Empowered New Media Data Analysis and Application Course [J]. Journal of Education and Educational Research, 2025, 14 (1): 109-112.
3. Zhou Y, Yang X, Xu W, et al. Rag GTPases control lysosomal acidification by regulating v-ATPase assembly in Drosophila. [J]. The Journal of biological chemistry, 2025, 301 (7): 110400.
4. Yang W, Sun Y, Yang F, et al. Preparation and properties of slow-release fertilizer containing urea encapsulated by pinecone biochar and cellulose acetate. [J]. International journal of biological macromolecules, 2025, 315 (P2): 144448.
5. He Y, Zhu X, Li D, et al. Enhancing Large Language Models for Specialized Domains: A Two-Stage Framework with Parameter-Sensitive LoRA Fine-Tuning and Chain-of-Thought RAG [J]. Electronics, 2025, 14 (10): 1961-1961.
6. Zhong X, Bai J, Deng C, et al. Pinecone-Structured ZnO microparticle coatings: A superhydrophobic approach for scale prevention on steel surfaces [J]. Surface Engineering, 2025, 41 (4): 463-470.
7. He X, Islam A M, Zhao T, et al. KOH‐Activated Pinecone Biochar for Efficient Chloramphenicol Removal From Aqueous Solutions [J]. CleanMat, 2025, 2 (1): 72-84.
8. Haowei Yang, Yu Tian, Zhongheng Yang, Zhao Wang, Chengrui Zhou, and Dannier Li. 2025. Research on Model Parallelism and Data Parallelism Optimization Methods in Large Language Model-Based Recommendation Systems. arXiv preprint arXiv:2506.17551. https://doi.org/10.48550/arXiv.2506.17551.
9. Huang L, Lu H. Design of intelligent financial data management system based on higher-order hybrid clustering algorithm. [J]. PeerJ. Computer science, 2024, 10 e1799.
10. Feiyun Sha, Changxu Ding, Xiaoyu Zheng, Jun Wang, and Yafang Tao. 2025. Weathering the Policy Storm: How Trade Uncertainty Shapes Firm Financial Performance through Innovation and Operations. International Review of Economics & Finance 102 (2025), 104274. https://doi.org/10.1016/j.iref.2025.104274.
11. Zhongheng Yang, Aijia Sun, Yushang Zhao, Yinuo Yang, Dannier Li, and Chengrui Zhou. 2025. RLHF Fine-Tuning of LLMs for Alignment with Implicit User Feedback in Conversational Recommenders. arXiv preprint arXiv:2508.05289. https://doi.org/10.48550/arXiv.2508.05289.
12. Feiyun Sha, Changxu Ding, Xiaoyu Zheng, Jun Wang, and Yafang Tao. 2025. Weathering the Policy Storm: How Trade Uncertainty Shapes Firm Financial Performance through Innovation and Operations. International Review of Economics & Finance (2025), 104274. https://doi.org/10.1016/j.iref.2025.104274.
13. Feiyun Sha, Jiawei Meng, Xiaoyu Zheng, and Yaqi Jiang. 2025. Sustainability Under Fire: How China-US Tensions Impact Corporate ESG Performance?. Finance Research Letters (2025), 107882. https://doi.org/10.1016/j.frl.2025.107882.
14. Xiaoyu Deng. 2025. Cooperative Optimization Strategies for Data Collection and Machine Learning in Large-Scale Distributed Systems. In Proceedings of the 2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT '25), Xi'an, China, 2025. IEEE, Piscataway, NJ, USA, 2151–2154. https://doi.org/10.1109/ISCAIT64916.2025.11010291
15. Jing Yang, Yuangui Wu, Yuping Yuan, Haozhong Xue, Sami Bourouis, Mahmoud Abdel-Salam, Sunil Prajapat and Lip Yee Por. 2025. LLm-AE-MP: Web attack detection using a large language model with autoencoder and multilayer perceptron. Expert Systems with Applications 274 (2025), 126982. https://doi.org/10.1016/j.eswa.2025.116982.

 doi:10.20944/preprints202510.1197.v1

16. Wei Yang, Yuzhen Lin, Haozhong Xue, and Jun Wang. 2025. Research on stock market sentiment analysis and prediction method based on convolutional neural network. In Proceedings of the 2025 International Conference on Machine Learning and Neural Networks (MLNN '25). Association for Computing Machinery, New York, NY, USA, 91-96. https://doi.org/10.1145/3747227.3747241
17. Wei Yang, Bochen Zhang, and Jun Wang. 2025. Research on AI economic cycle prediction method based on big data. In Proceedings of the 2025 International Conference on Digital Economy and Intelligent Computing (DEIC '25). Association for Computing Machinery, New York, NY, USA, 13-17. https://doi.org/10.1145/3746972.3746975
18. Yuping Yuan and Haozhong Xue. 2025. Cross-Media Data Fusion and Intelligent Analytics Framework for Comprehensive Information Extraction and Value Mining. International Journal of Innovative Research in Computer Science and Technology 13, 1 (2025), 50-57.