Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

A Cloud-Native Multi-Agent Generative AI Framework for Demand Forecasting and Capacity Optimization using Vertex AI

Dipans Verma
PhD Scholar, Department of Mathematics Amity University Maharashtra
Mumbai, India

Anshul Tiwari
Student
Lokmanya Tilak College of Engineering
Mumbai University, India

Ganesh Randave
Research Scholar, Dept. of Applied Science and Humanities, MIT-ADT University Pune, India

Ansh Agrawal
B.Tech Student
Ajeenkya DY Patil University
Pune, India

Sandeep Kulkarni*
Assistant Professor
Ajeenkya DY Patil University
Pune, India

Saksham Koli
B.Tech Student
Amity University Maharashtra
Mumbai, India

Priyanshu Kumar Sharma
B. Tech Student
Ajeenkya DY Patil University
Pune, India

***Abstract*—Demand forecasting and capacity planning is a challenge at enterprise level that demands intelligent, flexible and scalable solutions that can grow with the business. The traditional models don't work well, as they just do not provide a big picture and are too slow for making decisions. The paper introduces a new, cloud-native framework using Generative AI (GenAI), Retrieval-Augmented Generation (RAG) and Google Cloud Vertex AI to provide more intelligent demand forecasts and more precise resource planning. It combines a group of specialized AI agents dedicated to forecasting, analytics, and decision-making, with microservices ensuring elastic scaling. A practical example of real business environment shows the advantages achieved in forecasting accuracy, resource utilization, and decision-making speed.**

***Index Terms*—Generative Artificial Intelligence, Large Language Models, Multi-Agent Systems, Demand Forecasting, Capacity Optimization, Retrieval-Augmented Generation, Cloud-Native Architecture, Vertex AI, Hybrid Forecasting, Decision Intelligence**

I. INTRODUCTION

Demand forecasting and capacity planning are crucial for sound business decision-making, optimizing resource use, improving efficiency, and aligning strategies with market demand. Conventional machine learning techniques and classical statistical models work well when the data is clean and structured, but the same does not apply to real enterprise environments; here, unstructured information, changing conditions, and streaming data are commonplace and are not well addressed by older methods.

With the arrival of Large Language Models (LLMs) and cloud-based AI platforms, agent-based systems that can reason, plan, and execute complex workflows have emerged.Large Language Models (LLMs) and cloud-based AI platforms have ushered in an era of agent-based systems that can reason, plan, and execute complex workflows [15]. These systems have the ability to understand context, to access outside information and to adjust their decisions on the fly. At the heart of this change is the addition of Retrieval-Augmented Generation (RAG): RAG significantly enhances the factual correctness and relevance of LLM-based outputs by enriching them with recent, external data [9].

In the case of cloud platforms like Google Cloud Vertex AI, this becomes possible on a large scale, with managed model orchestration, real-time inferences, and enterprise data integration in one place [6]. The integration of LLMs, RAG, and scalable cloud infrastructure presents a promising opportunity for enterprise forecasting and decision-making to become more intelligent and transformative.

The paper presents a multi-agent, RAG-enabled cloud-native approach to enhance demand forecasting and capacity planning within Vertex AI. The system uses dedicated agents

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

for forecasting, optimization, analytics and reporting, allowing the system to be modular and easily scalable or customizable. Hybrid retrieval – semantic search and recommendation-based ranking – maintains relevance and accuracy of the predictions even in the face of changing data.

II. CONTRIBUTIONS

This work makes the following contributions:

- A scalable multi-agent GenAI system for demand forecasting and capacity planning, facilitating modular and distributed decision-making, leveraging recent develop- ments in agentic systems and foundation models.Scalable multi-agent GenAI system for demand forecasting and capacity planning, enabling modular and distributed decision-making aligned with advances in agentic sys- tems and foundation models.
- A Retrieval-Augmented Generation (RAG) system built on Vertex AI that brings enterprise-specific knowledge directly into forecasting and analytics, increasing the grounding and accuracy of the content.
- A hybrid forecasting method combining classical statisti- cal time-series forecasting with the reasoning from LLMs under changing and uncertain conditions [2].
- A personalized retrieval and ranking system that com- bines semantic search, keyword based retrieval (BM25) and collaborative filtering signals to ensure the results are relevant and user adapted.
- An empirical assessment in a real enterprise context that shows better forecasting performance, more intelligent resource utilization, and better decision making at large workflow sizes.

III. SYSTEM ARCHITECTURE

A. Overview

The framework is deployed on Google Cloud Platform and is distributed and cloud-native with multiple agents. They are all implemented as microservices that communicate through event-driven pipelines, enabling scalability and flexibility to accommodate growing workloads.

The architecture revolves around the concept of using RAG, Hybrid retrieval (semantic + keyword search), and LLM reasoning to generate personalized and contextually relevant outputs. Agent orchestration integrates forecasting, optimization, analytics and reporting into a seamless workflow, but without compromising speed or reliability for enterprise class operations.

B. Agent Design

Every agent has a well-staked-out position in the pipeline: Sets forecasting goals, chooses data sources, and directs tasks to the right agents as per workflow orchestration techniques, **Planner Agent:** starts the forecasting process.

- **Forecasting Agent**: Generates forecasts using statistical time series analysis coupled with LLM-based inference and delivers both accuracy and context.
- **Optimization Agent**: Manages capacity planning, reducing the difference between forecasted demand and resources available and optimising the use of resources, at the enterprise level.
- **Analytics Agent:** Keeps an eye on KPIs, assess forecasting performance and gives continuous feedback to enable iterative improvement.
- **Reporting Agent**: Uses LLMs to create concise, digestible reports, summaries, and actionable insights for decision-makers.

C. Cloud Architecture (GCP)

The system is deployed using a cloud-based, managed GCP stack:

- **Vertex AI:** This is the main AI engine where you can deploy your models, perform LLM inference (Gemini), manage features and manage end-to-end pipelines [6].
- **Cloud Run and GKE:** Scale, high availability, and manage the workload for containerized microservices.
- **Pub/Sub:** Asynchronous messaging backbone which gives agents the ability to communicate and share data, without being coupled. BigQuery: Enterprise data warehouse for historical and real-time data storage, feature engineering and large-scale analytics [11].

D. End-to-End Workflow

Consume enterprise data, historical and real-time.

1) Planner Agent sets forecasting goals and triggers the workflow.
2) The hybrid model is used for forecasting Agent's demand.
3) RAG component gathers relevant enterprise knowledge to provide grounding context.
4) Optimization Agent automatically distributes resources depending on forecasting results.
5) Analytics Agent assesses the KPIs and monitors the system performance.
6) Reporting Agent: Clear, Actionable insights with LLM-powered language generation.

IV. METHODOLOGY

A. Hybrid Forecasting Model

To enhance robustness and contextual awareness, a hybrid forecasting strategy combines traditional statistical time-series models with LLM-based predictions:

$$D_t = \alpha D_{\text{stat}} + (1 - \alpha) D_{\text{llm}} \quad (1)$$

where D_{stat} represents outputs from statistical models (e.g., ARIMA, regression) [2], and D_{llm} denotes predictions generated by LLMs deployed on Vertex AI [4]. The weighting parameter $\alpha \in [0, 1]$ is optimized empirically.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

B. RAG-Based Contextual Retrieval

To incorporate enterprise knowledge and improve contextual reasoning, a hybrid RAG mechanism is adopted:

$$R_{\text{hybrid}} = \beta R_{\text{dense}} + (1 - \beta) R_{\text{sparse}} \tag{2}$$

where R_{dense} denotes embedding-based semantic retrieval and R_{sparse} denotes keyword-based retrieval (e.g., BM25) [9], [14]. The parameter β controls the trade-off between semantic relevance and lexical precision.

C. Two-Tower Semantic Retrieval

To enable scalable semantic matching, a two-tower (dual encoder) architecture is employed:

$$S_{\text{semantic}}(q, d) = \frac{f_q(q) \cdot f_d(d)}{\|f_q(q)\| \, \|f_d(d)\|} \tag{3}$$

where $f_q(\cdot)$ and $f_d(\cdot)$ represent query and document encoders, respectively. This formulation enables efficient approximate nearest neighbor (ANN) retrieval at large scale [7].

D. Personalized Retrieval and Ranking

To incorporate user preferences, a recommendation-aware ranking mechanism is integrated:

$$S_{\text{final}} = w_1 S_{\text{semantic}} + w_2 S_{\text{keyword}} + w_3 S_{\text{rec}} \tag{4}$$

where S_{rec} is derived from collaborative filtering models (e.g., LightFM), capturing user interaction patterns [13]. The weights w_1, w_2, w_3 are tuned to balance relevance and personalization.

E. Optimization Objective

The capacity planning problem is formulated as:

$$C_t^* = \arg\min_{C_t} \sum_t (D_t - C_t)^2 \tag{5}$$

This objective minimizes demand-supply mismatch and ensures efficient resource utilization under operational constraints [3].

F. Training Strategy

The two-tower model is trained using contrastive learning:

$$\mathrm{L} = -\log \frac{\exp(S(q, d^+))}{\exp(S(q, d^+)) + \exp(S(q, d^-))} \tag{6}$$

where d^+ and d^- denote positive and negative samples. This loss improves retrieval accuracy in large-scale settings [5].

G. End-to-End Pipeline

The framework integrates forecasting, retrieval, personalization, and optimization into a unified, scalable pipeline:

1) **Hybrid Forecasting:** Predicts demand by combining classical time-series models with LLM-powered contextual reasoning.
2) **Contextual Retrieval (RAG):** Blends dense (embedding-based) and sparse (keyword-based) search to retrieve the most relevant enterprise knowledge.
3) **Semantic Matching:** Uses a dual-encoder setup to assess query-document similarity.
4) **Optimization:** Allocates resources to minimize demand-capacity mismatches.
5) **Insight Generation:** Produces clear, actionable summaries in natural language using LLMs.

Algorithm 1 Multi-Agent Demand Planning and Decision Framework

Require: Historical data H, real-time signals S, user context U

Ensure: Demand forecast D_t, capacity plan C_t, insights I

Initialize Planner Agent and define forecasting objectives
Preprocess data: $H, S \rightarrow F$
/* **Forecasting Phase** */
$D_{\text{stat}} \leftarrow f_{\text{stat}}(F)$
$D_{\text{llm}} \leftarrow f_{\text{llm}}(F, U)$
$D_t \leftarrow \alpha D_{\text{stat}} + (1 - \alpha) D_{\text{llm}}$
/* **Retrieval Phase** */
$\mathrm{D} \leftarrow$ HybridRetrieve(dense, sparse)
$S_{\text{semantic}}(q, d) \leftarrow \frac{f_q(q) \cdot f_d(d)}{\|f_q(q)\| \|f_d(d)\|}$
$S_{\text{rec}} \leftarrow f_{\text{CF}}(U, \mathrm{D})$
$S_{\text{final}} \leftarrow w_1 S_{\text{semantic}} + w_2 S_{\text{keyword}} + w_3 S_{\text{rec}}$
$\mathrm{D}_K \leftarrow$ TopK($\mathrm{D}, S_{\text{final}}$)
/* **Optimization Phase** */
$C_t \leftarrow \arg\min_C \sum_t (D_t - C_t)^2$
/* **Analytics and Reporting** */
Evaluate KPIs and forecasting errors
$I \leftarrow f_{\text{llm}}(\mathrm{D}_K, D_t, C_t)$
return D_t, C_t, I

V. Case Study: Enterprise GenAI Demand Planning System

A. Project Overview

A practical GenAI solution for Demand Forecasting and Capacity Planning was built and implemented to test the proposed framework in a real-world scenario. It is designed to support high-volume automation and analytics workflows and includes multi-agent orchestration, RAG, and reasoning powered by LLM to provide context-aware, data-driven decision intelligence.

B. Implementation Details

The system deployed will contain the following:

- Modular, scalable workflows are orchestrated by Multi-Agent Orchestration (MAO) using the ADK and Model Context Protocol (MCP).
- Google Cloud Vertex AI (Gemini) provides contextual reasoning, forecasting, and natural language insight generation via Google's LLM integration.
- Vector-based search (FAISS, ChromaDB) with keyword-based search (BM25) for better recall and precision.Hybrid Retrieval Pipeline: Use vector-based search (FAISS, ChromaDB) and keyword-based search (BM25) to improve recall and precision. [9], [14]

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

- Some recommendation models like LightFM [13] add a Personalization layer that takes user interaction data to make the ranking results more relevant.
- **Data Pipeline:** Pub/Sub-based real-time event-driven data ingestion and processing for enterprise data.
- **Data Storage and Analytics:** BigQuery can store large amounts of data, feature engineer and perform analytics [11].
- **High availability, resilience and elastic scaling made easy with containerized microservices on Cloud Run and GKE.**

C. Use Cases

The following enterprise scenarios were tackled by the system:

- Predicting customer needs for AI and automation systems in volatile and uncertain market conditions.
- Optimize capacity and resources to achieve better utilisation and minimise inefficiencies.
- Financial modeling, ROI analysis with predictive analytics and AI-informed insights.
- Real-time dashboards and executive summaries created using LLM.

D. Business Impact

Deployment of the framework delivered the following measurable outcomes:

- Manual planning effort reduced by 35-40
- Contextual and real-time signals were utilized to enhance the accuracy of the forecasts.
- Resource utilization increased with smart allocation of resources.
- Automated insights and real-time analytics sped up decision cycles.

VI. DISCUSSION

The findings highlight the power of using multi-agent architectures with cloud-native AI environments such as Vertex AI to tackle enterprise demand planning and decision intelligence problems. There are a number of benefits.

The hybrid forecasting model combines statistical time-series analysis with LLM-generated reasoning, resulting in more context-sensitive forecasts that are robust enough to adapt to evolving business conditions. The combination of RAG can leverage the domain knowledge of the system, significantly enhancing its accuracy and interpretability [9].

The newly introduced personalized recommendation signal leads to more salient information when retrieved by two-tower architectures, which are based on the semantic retrieval. Following the recent developments of neural retrieval and collaborative filtering [13], [14], the multi-stage retrieval and ranking process considers both contextual similarity and user preferences.

The system is designed as a cloud-native application that leverages Vertex AI, containerized microservices, and event-driven components like Pub/Sub, making it easily scalable to handle high volumes of data and real-time demands.

Several limitations remain. Latency comes from retrieval, ranking, and inference of LLMs, which necessitates approximate nearest neighbor search and model optimization to ensure real-time performance. LLM calls can be frequent, and cloud resource usage can be high – costs need to be reduced by intelligent scheduling. If you rely on the stable and consistent output of your LLM, you need guardrails, monitoring and constant evaluation. Last, but most important, is the importance of data quality: forecasts and personalisation suffer if input data is out of date or incomplete, and needs to be monitored and continually re-trained.

VII. CONCLUSION AND FUTURE WORK

This paper introduces a cloud-native, multi-agent Generative AI platform for enterprise demand forecasting and capacity planning. The framework combines the power of hybrid forecasting, RAG, two-tower semantic retrieval, and recommendation-based ranking, all within a single system, enabling context and personalization to be addressed at scale. Google Cloud Vertex AI offers the managed infrastructure for quick deployment and inference of models.

The use of statistical models along with reasoning techniques based on LLM enhances robustness in the face of changing business requirements, and layered retrieval and personalization further enhance accuracy and relevance. The framework is evaluated empirically and shown to be more accurate and less resource-intensive than traditional and conventional machine learning techniques, confirming the usefulness of combining predictive analytics, contextual knowledge retrieval and personalisation for users in a multi-agent design.

Future directions encompass: reinforcement learning for adaptive, self-optimizing agents; autonomous self-orchestrating agents for closed-loop decision making; extension to multi-cloud and edge environments to minimize latency; cost saving due to smarter caching and selective invocation of LLMs; explainability methods to create enterprise trust; and continuous learning methods to address data drift in dynamic environments.

It offers a hands-on overview of the new generation of enterprise decision intelligence, that's making use of trusted forecasting techniques and the adaptability of modern Generative AI.

ACKNOWLEDGMENT

The authors thank Ajeenkya DY Patil University, Amity University Maharashtra, and MIT-ADT University for providing the collaborative environments and resources that supported this research.

REFERENCES

[1] R. Bommasani et al., "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.
[2] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ: Wiley, 2015.
[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.
[4] T. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[5] T. Chen et al., "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, 2020.
[6] Google Cloud, "Google Cloud Vertex AI Documentation," 2023. [Online]. Available: https://cloud.google.com/vertex-ai
[7] M. Henderson et al., "Efficient natural language response suggestion for smart reply," *arXiv preprint arXiv:1705.00652*, 2017.
[8] J. Kreps, "Kafka: A distributed messaging system," 2011.
[9] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[10] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: Results, findings, conclusion and way forward," *International Journal of Forecasting*, 2018.
[11] S. Melnik et al., "Dremel: Interactive analysis of web-scale datasets," in *Proc. VLDB*, 2010.
[12] S. Newman, *Building Microservices*. Sebastopol, CA: O'Reilly, 2015.
[13] S. Rendle et al., "Neural collaborative filtering vs. matrix factorization revisited," in *Proc. RecSys*, 2020.
[14] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, 2009.
[15] M. Wooldridge, *An Introduction to MultiAgent Systems*. Hoboken, NJ: Wiley, 2009.
[16] G. Zhang et al., "Deep learning for time series forecasting: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)