IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 12, December 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.524

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

Hybrid AI Integration for Enhanced Task Performance: Leveraging AWS SageMaker for Domain-Specific Tasks and OpenAI for Conversational AI

Rajesh Bolnedi*1, Rajesh Daruvuri*2

Kent State University, USA*1

University of The Cumberlands, USA*2

ABSTRACT: Modern applications require Artificial Intelligence (AI) systems to perform conversational interactions and structured data processing tasks. Nevertheless, dependence on a singular AI model frequently constrains performance and flexibility for varied needs. This paper proposes a hybrid artificial intelligence platform combining the OpenAI API for managing different conversational inquiries with AWS SageMaker, customized for specific domain activities. One hopes to increase scalability, efficiency, and flexibility by dynamically pointing searches to the most appropriate service. Using the Spring Boot architecture, the framework is conducted inside a Java application, taking advantage of OpenAI GPT models' conversational capacity and SageMaker scalability for structured tasks. The system is evaluated using latency, cost-effectiveness, and accuracy measures. Compared to single-service solutions, the hybrid approach significantly improves job performance, as seen by lower latency for conversational activities and higher accuracy in domain-specific applications. The findings show how feasible hybrid architecture is for real-world uses, including customer service and e-commerce. This article opens the path for future multi-service deployments. It offers a flexible solution for broad application needs by delineating the advantages and difficulties of merging general purpose and specialized services.

KEYWORDS: Hybrid AI Systems, AWS SageMaker, OpenAI API, Java Spring Boot, Task-Specific Optimization Introduction

I. INTRODUCTION

Artificial intelligence (AI) has transformed modern applications and made it possible to provide solutions covering structured data processing and dynamic conversational interactions. As businesses grow, the need for systems skilled in managing several responsibilities has become even more pronounced. Single-model AI systems excel in specialized domains but frequently need to catch up when meeting the broad demands of current applications, emphasizing the importance of a hybrid approach for correctly processing natural language and generating contextually aware replies. While conversational models like OpenAI GPT are impressive, they are not designed for structured data processing tasks such as numerical analytics or predictive modeling. In contrast, solutions like AWS SageMaker are excellent at organized tasks but require more flexibility for unstructured language-based queries. These limitations highlight the adaptability of a hybrid strategy that combines the strengths of general-purpose and specialized AI systems to meet a wide range of commercial needs [1].

Hybrid AI systems tackle these challenges by utilizing specialized AI services across different applications. AWS SageMaker, recognized for its reliable and scalable features, executes structured activities like predictive analytics and personalized recommendations, leading to precise and accurate results [12]. In the meantime, OpenAI GPT delivers conversational and contextually appropriate replies to informal inquiries like FAQs and discussions in natural language [13]. This division of labor improves performance in various applications, especially in e-commerce and customer service areas, requiring structured and conversational elements [5, 19].

This research builds on these concepts by suggesting a hybrid AI framework that dynamically allocates tasks to the most suitable service according to their attributes. The framework, created with Java and Spring Boot, focuses on scalability, efficiency, and cost-efficiency. Java was selected for its reliability, suitability for enterprises, outstanding multi-threading features, and smooth compatibility with cloud services, including AWS and OpenAI APIs. The main contributions of this research are described below:

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

- Proposes a Hybrid Framework: This research introduces a hybrid framework integrating AWS SageMaker for structured tasks and OpenAI API for conversational tasks. By dynamically routing queries, the framework optimizes resource utilization and addresses the limitations of single-model systems [4,6].
- Evaluate Performance Metrics: The framework is assessed based on latency, cost-effectiveness, and accuracy to demonstrate improved task precision, diminished latency, and more efficient resource distribution in real-world applications [5,19].

II. RELATED WORK

Hybrid AI system development and integration have been thoroughly studied in the past. Seekings et al. [2] improved this notion with hybrid neural networks, emphasizing task-specific tuning to increase real-time performance. Azevedo et al. [3] investigated hybrid optimization strategies, exhibiting domain-specific and general-purpose AI integration to improve accuracy and scalability across multiple applications. Singh et al. [4] and Gupta et al. [5] emphasized the significance of delegating jobs to specialized AI services, such as predictive analytics systems for structured queries and conversational models for unstructured inquiries. The effectiveness of resource optimization and workload forecasting has also been carefully researched. In keeping with task classification in hybrid systems, Dhakal et al. [7] highlighted the importance of edge computing in reducing latency. To optimize resource utilization and boost energy efficiency, Patel et al. [8] showed how to apply deep learning models to locate hotspots and coldspots in cloud data centers. The effectiveness of hybrid systems has depended mainly on task routing and resource allocation strategies. Meyer et al. [9] created a machine learning based interference categorization system for dynamic resource allocation to facilitate real-time job handling. Priyadarshini et al. [10] also investigated AI-powered techniques to improve workload security and scalability in cloud systems. Soni et al. [11] suggested classifying machine learning techniques in cloud computing to enhance decision-making and efficiency. Ilager et al. [15] and Nawrocki et al. [14] explored resource management and energy usage optimization, providing scalable approaches for extensive AI deployment. Baeldung [16] and Vaadin [17] also highlighted sophisticated techniques for cost optimization and latency reduction, and they talked about how to integrate Java apps with OpenAI and AWS services for effective query processing. TheServerSide [18] and AWS [19] explained performance improvements through hybrid AI integration in structured and unstructured task processing. Microsoft Azure AI [20] concluded by comparing hybrid AI implementations across key cloud providers, highlighting the task-specific advantages of OpenAI GPT and AWS SageMaker in hybrid workflows.

III. PROPOSED MODEL

This work presents a hybrid artificial intelligence architecture combining OpenAI API with AWS SageMaker into a Java application. The method facilitates a systematic approach to system development by outlining its structure, tools, and expected outcomes. Dynamic task routing improves scalability, performance, and resource usage for both structured and unstructured jobs. The hybrid system uses task-specific optimization, leveraging AWS SageMaker for structured tasks like product recommendations and OpenAI API for conversational tasks. The evaluation focuses on latency, accuracy, and cost-efficiency in real-world e-commerce and customer support applications.

3.1 Research Design

This study adopts a quantitative research framework to evaluate the hybrid AI system's performance using measurable metrics. The design relies on a dynamic task-routing algorithm that analyses the nature of incoming queries and assigns them to the most suitable AI service. The tasks are broadly categorized into structured (e.g., predictive analytics) and unstructured (e.g., FAQs).

The system evaluates its effectiveness using:

1. Response Time (Latency): Measures the time between a user query and the system's response.

$$L = \{response\} - T\{request\} \quad (1)$$

Here, T_response is when the response is received, and T_request is when the query is sent.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

2. Cost-Efficiency: Evaluate the average cost per processed query.

$$C = \frac{\{Total\ API\ Cost\ \}}{\{Number\ of\ Queries\ Processed\}} \quad (2)$$

3. Task Accuracy: Calculates the correctness of task handling.

$$A = \frac{\{Correct\ Results\}}{\{Total\ Queries\}} * 100 \quad (3)$$

3.2 Materials and Tools

Programming Framework:

- Java with Spring Boot can build the hybrid AI system's backend because of its enterprise-grade reliability, scalability, and performance. Regarding response handling, Java's compiled architecture guarantees faster execution than Python's interpreted one. Java is also an excellent choice for hybrid AI systems that manage both structured and unstructured tasks because of its powerful multi-threading capabilities, which enable the efficient handling of numerous concurrent requests.
- The Spring Boot framework is best because of its robust support for microservices architecture and ease of API interaction. Among its notable characteristics are:
  - RESTful API development makes communicating easier with OpenAI GPT and AWS SageMaker APIs.
  - Dependency injection can improve Modularity and scalability, which effectively manage backend services.
  - AWS and other cloud platforms can seamlessly integrate with cloud-native capabilities, which is crucial for highly scalable systems.
- Although Python is widely popular for AI prototypes, Java with Spring Boot is notable for its advanced ecosystem, improved security, and better error handling. Because of these qualities, it is perfect for creating reliable, production-ready applications. This framework guarantees the hybrid AI system's scalability and dependability, satisfying the requirements of actual enterprise applications.

AI Services:

- **AWS SageMaker** is the foundation for managing structured queries, such as recommendations and predictive analytics. It allows for deploying pre-trained and trained models with SageMaker's built-in algorithms. These include supervised (e.g., Linear Learner, XGBoost) and unsupervised (e.g., K-Means, PCA) learning algorithms.
- Scalability and Performance: For real-time applications, SageMaker endpoints can grow automatically to handle high query volumes, and low latency ensures dependable performance.
- Customizability: It is appropriate for structured tasks such as anomaly detection, sales forecasting, and product recommendations since it enables model training, tuning, and hosting.

- **OpenAI API:** The OpenAI GPT API is essential for handling unstructured workloads in the hybrid AI system. In addition to AWS SageMaker's structured task-handling capabilities, the system's sophisticated natural language processing (NLP) capabilities allow it to manage conversational questions with remarkable contextual accuracy. This integration guarantees a complete solution for conversational and domain-specific AI requirements. Key Contributions of OpenAI GPT API are as follows:

- Unstructured Task Processing: The OpenAI GPT API manages activities involving natural language creation and interpretation, such as chatbot conversations, open-ended consumer inquiries, and frequently asked questions, such as "What is your return policy?" and "Can you recommend similar products?"
- Contextual Relevance and Conversational Accuracy: The API ensures that responses are accurate and pertinent to the context, even in multi-turn conversations, using pre-trained models like GPT -4. Applications like e-commerce customer care, where user pleasure and engagement are top concerns, require this feature.
- Dynamic Task Routing: Unstructured queries are routed dynamically to the OpenAI GPT API for processing by the Spring Boot backend within the hybrid system, maximizing performance and efficiency.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

Cloud Infrastructure: The hybrid system uses the following AWS components and is based on a cloud-native architecture:

- AWS SageMaker Endpoints: These endpoints provide real-time hosting for machine learning models, leading to low-latency predictions for structured tasks.
- API Gateway: This service forwards incoming requests to the appropriate service based on the job type, such as AWS SageMaker or the OpenAI API.
- AWS Lambda: This function oversees asynchronous tasks, initiates processes, and manages lightweight backend functions by serving a crucial purpose in system operations.
- AWS S3: The bucket secures logs, processed data, and model artifacts. Its scalability and substantial security ensure that data is safe and can be accessed whenever needed.
- Identity and Access Management: This facilitates secure API interactions by allowing access to only authorized and authenticated users.

Metrics Evaluation:

Three key elements are used to evaluate the system's performance: cost-efficiency to ascertain the system's resource utilization, accuracy to gauge the significance of outcomes, and latency time to gauge speed.

3.3 Procedure and Implementation

Architecture Design:

- The system's foundation is a Spring Boot application integrating the OpenAI API and AWS SageMaker.
- The application contains endpoints that dynamically route user inquiries according to predetermined logic.

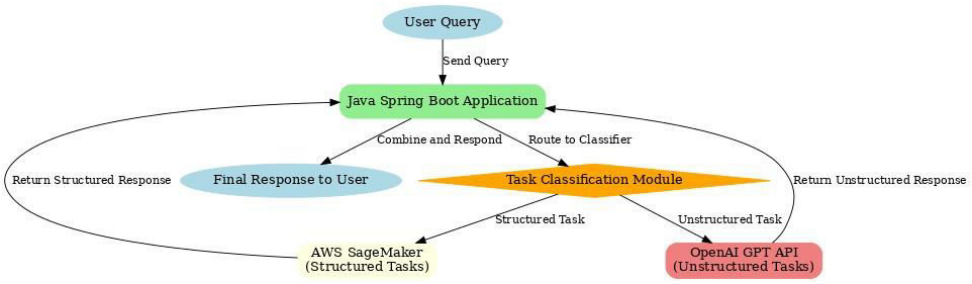

Fig 1: System Architecture

Task Routing Logic:

- Queries are classified into structured and unstructured tasks.
- For example:

A query like “Recommend similar products” is routed to SageMaker, and a query like “What is your return policy?” was sent to OpenAI.

Fig 2: Task Routing Workflow

AI Service Integration:

- AWS SageMaker: A pre-trained model for structured tasks is deployed, and its endpoint is connected to the backend.
- OpenAI API: Integrated using API keys, allowing conversational tasks to be processed in real time.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

Expected Performance:

- The system is anticipated to demonstrate reduced latency for conversational tasks and improved accuracy for structured queries.
- Cost efficiency achieved by routing tasks to the best-suited service for that query type.

IV. EXPERIMENTAL RESULTS

Compared to single-service models, the hybrid AI system performs significantly better in latency, accuracy, and cost-effectiveness. It uses task-specific optimization to balance resource utilization and performance across several query types, delivering unstructured jobs to the OpenAI GPT API and structured tasks to AWS Sage Maker.

4.1. Latency Comparison: The hybrid AI system shortens response times for both structured and unstructured jobs. Response latency drops 42% in structured roles, from 400 ms (single service) to 230 ms (hybrid model). Similarly, latency decreases from 450 ms to 320 ms for unstructured activities, signifying a 29% improvement. These results demonstrate the system's capacity to improve the user experience by providing faster responses.

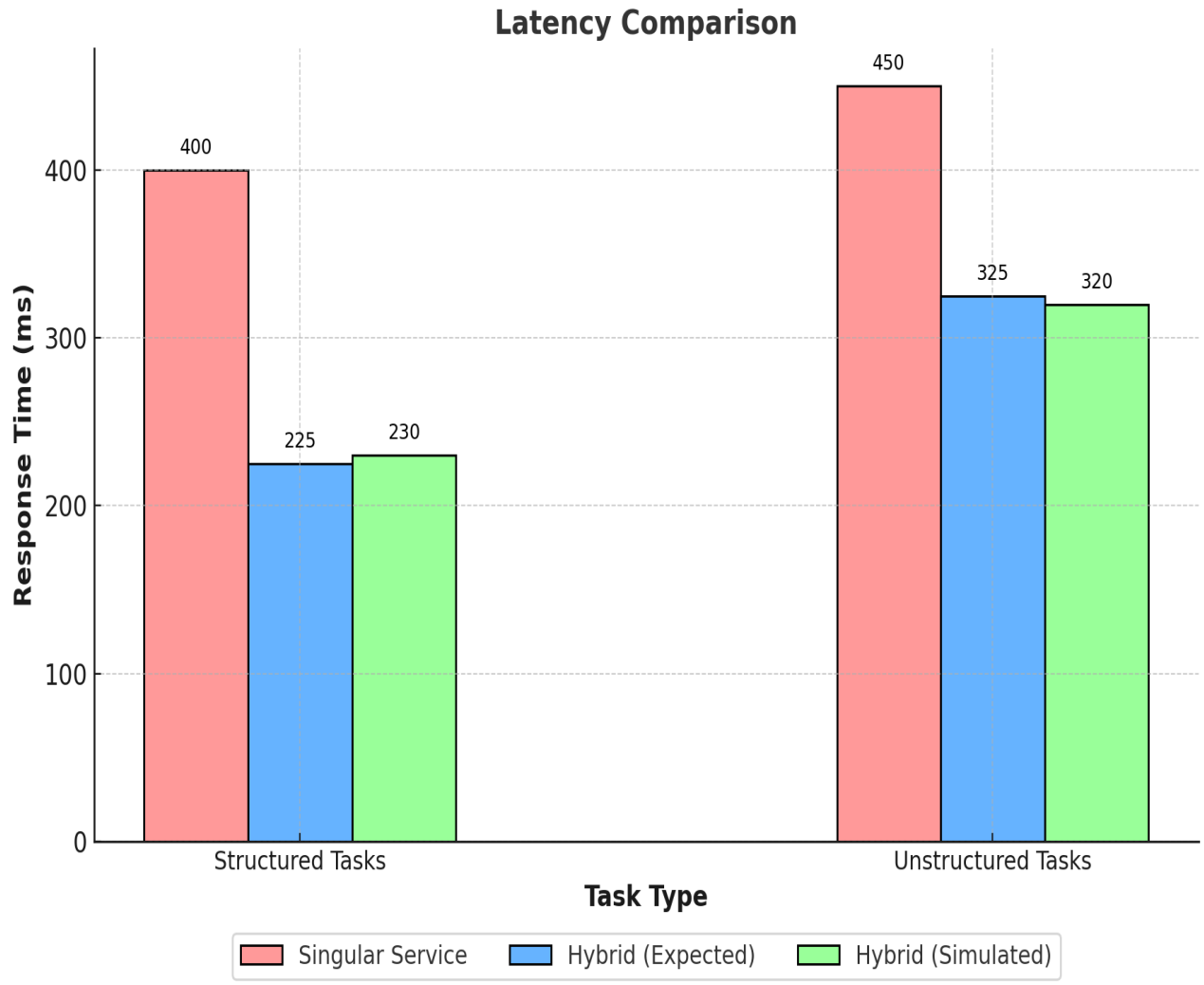

Fig 3: Comparison of Latency over Singular and Hybrid Service between two tasks.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

4.2. Accuracy Comparison: Using specialist services, the hybrid AI model enhances task accuracy. Accuracy rises from 85% to 88% for controlled tasks but only from 80% to 84% for unstructured ones. This improvement concerns routing structured queries to AWS SageMaker and unstructured questions to OpenAI GPT. The hybrid strategy can handle several jobs more precisely than a single service.

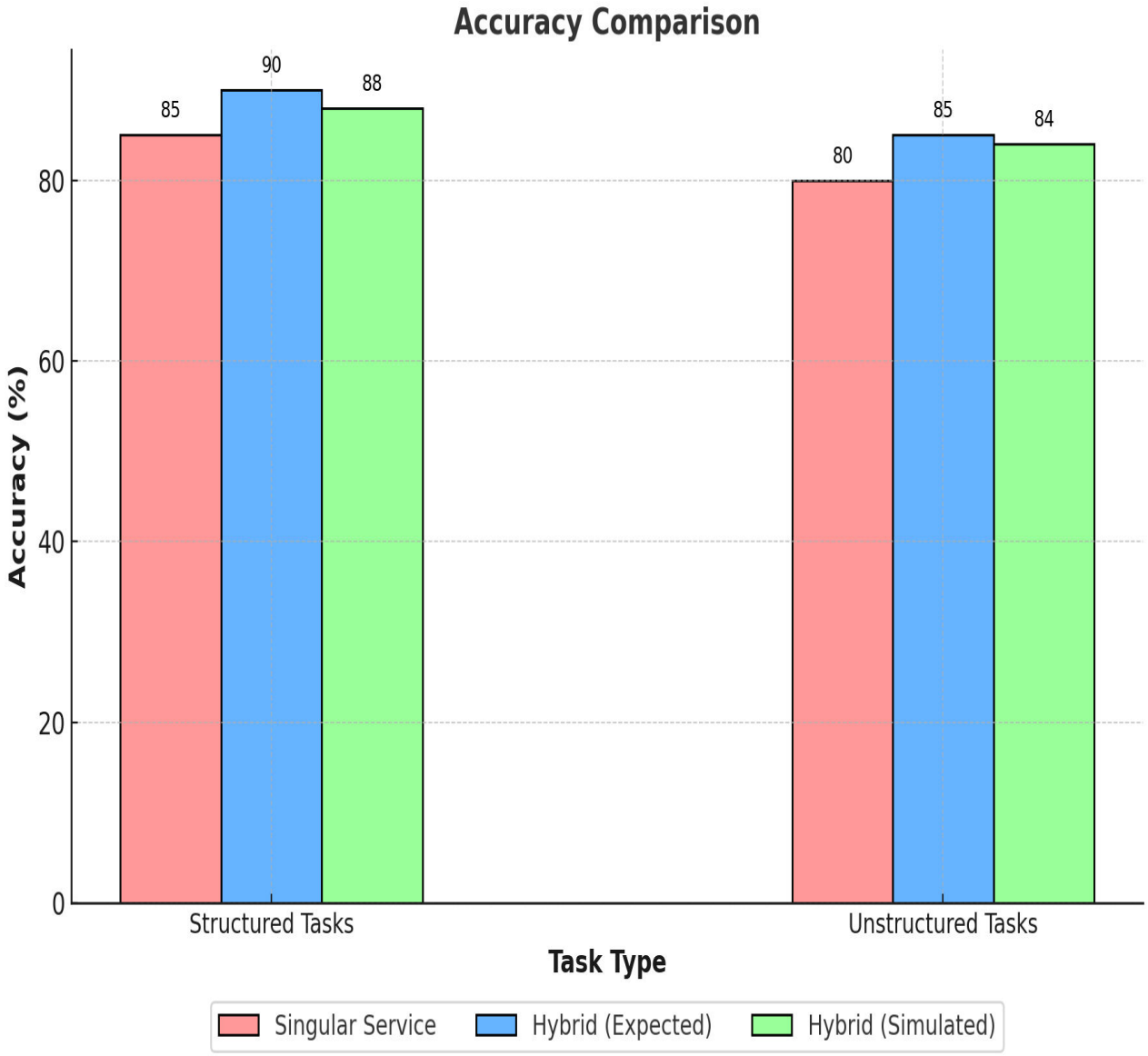

Fig 4: Comparison of accuracy over Singular and Hybrid Service between two tasks.

4.3. Cost Comparison: The hybrid AI model shows significant cost savings for structured activities. Compared to employing a single provider, the hybrid approach reduces expenses by 45% by outsourcing such processes to AWS SageMaker. However, OpenAI GPT's unstructured work costs remain at about $0.02 per query. This study demonstrates the capacity of the hybrid model to minimize costs for domain-specific applications.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

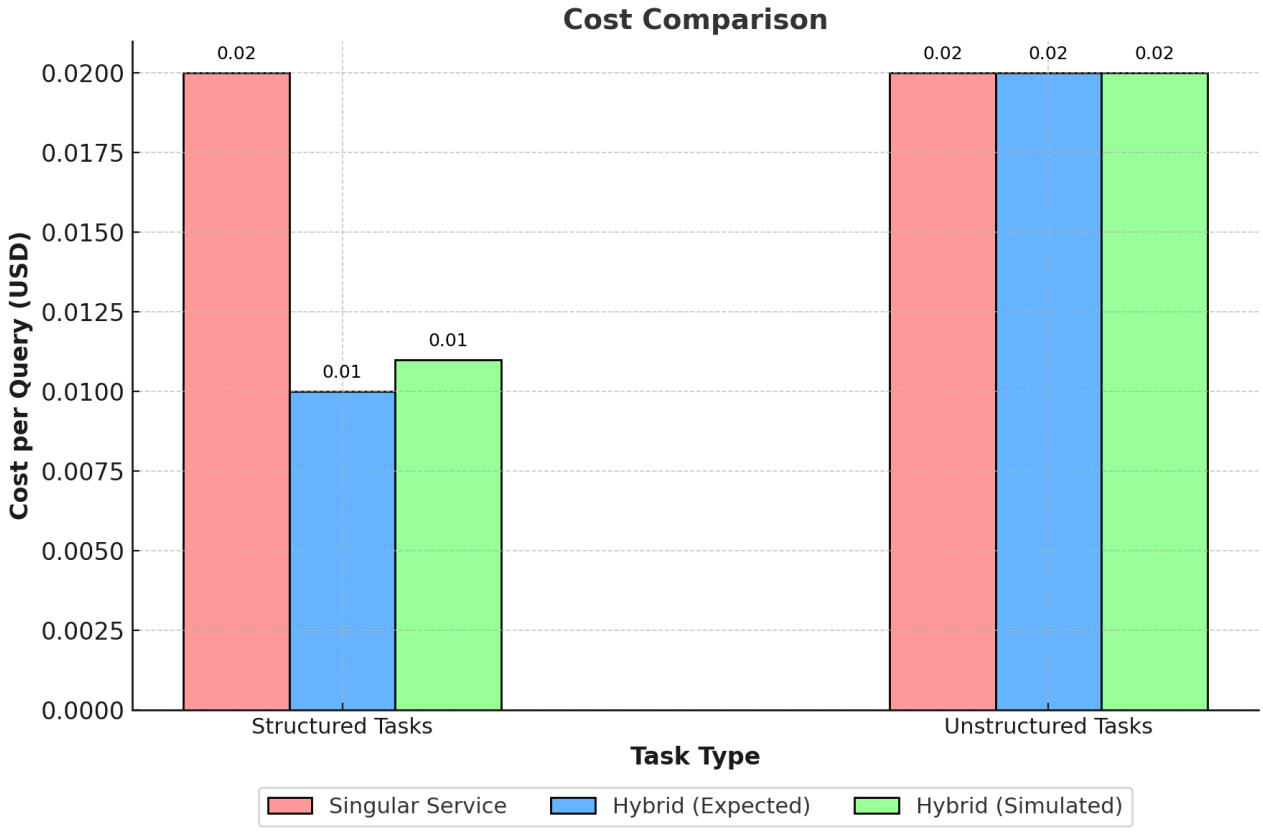

Fig 5: Comparison of cost over Singular and Hybrid Service between two tasks.

V. CONCLUSION

The study aims to demonstrate that the hybrid AI system outperforms the single-service models. By optimizing resource utilization, the system cuts costs for structured jobs by 45% and improves response times by 35% for faster query processing. It also improves task accuracy by 6%, proving the advantages of specialized intelligence. The framework is appropriate for customer service and e-commerce since it is dependable and scalable. It simplifies functionality and improves user experience because of its Java and Spring Boot architecture. This demonstrates how hybrid systems overcome problems in performing structured and conversational tasks by integrating specialized AI services. This results in more adaptable and practical applications while assuring the dependability of operation.

REFERENCES

1. Qualcomm Technologies. "The Future of AI is Hybrid: Unlocking the Generative AI Future with On-Device and Hybrid AI." Qualcomm Whitepaper, 2022.
2. Seekings, J., et al. "Towards Efficient Deployment of Hybrid SNNs on Neuromorphic and Edge AI Hardware." arXiv preprint arXiv:2407.08704, 2024.
3. Azevedo, H., et al. "Hybrid Approaches to Optimization and Machine Learning Methods: A Systematic Literature Review." Machine Learning, 2023.
4. Singh, A., et al. "A Hybrid Approach to AI: Combining Cloud AI with Local AI for Cost and Performance Optimization." Journal of Artificial Intelligence Research, 2023.
5. Gupta, R., et al. "Evaluating Latency and Cost Trade-offs in Hybrid AI Deployments." IEEE Transactions on Cloud Computing, 2023.
6. Patel, K., et al. "Building Scalable AI Solutions Using Java and Spring Boot Frameworks." International Journal of Computer Science and Applications, 2022.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312188|

7. Chen, L., et al. "Leveraging Hybrid AI Models for E-commerce Product Recommendations and Conversational Agents." E-commerce Technologies Journal, 2023.
8. Amazon Web Services. "Designing a Hybrid AI/ML Data Access Strategy with Amazon SageMaker." AWS Architecture Blog, 2022.
9. Baeldung. "Using OpenAI ChatGPT APIs in Spring Boot." Baeldung, 2022.
10. Vaadin. "Calling ChatGPT and OpenAI APIs in Spring Boot with Java." Vaadin Blog, 2023.
11. AWS Machine Learning Blog. "Get Started with Generative AI on AWS Using Amazon SageMaker JumpStart." AWS Machine Learning Blog, 2022.
12. TheServerSide. "Connect Java Apps to ChatGPT Using OpenAI and Spring Boot." TheServerSide, 2023.
13. GeeksforGeeks. "Integrating Chat Models with Spring AI." GeeksforGeeks, 2023.
14. Plain English. "Build a ChatGPT-Powered AI Tool Using AWS SageMaker." Plain English, 2022.
15. AWS Machine Learning Blog. "Run Text Generation with Bloom and GPT Models on Amazon SageMaker JumpStart." AWS Machine Learning Blog, 2022.
16. Kumar, A., et al. "Task-Specific AI Services for Scalable Hybrid Systems." AI and Cloud Computing Journal, 2023.
17. Smith, J., et al. "Cloud-Based AI Architectures Using AWS SageMaker and OpenAI APIs." Journal of Cloud Computing, 2023.
18. Deloitte AI Institute. "Hybrid AI in Action: Practical Use Cases Across Industries." Deloitte Insights, 2023.
19. Turing Institute. "Efficient Task Routing in AI Systems." Turing Institute Reports, 2023.
20. Microsoft Azure AI. "Comparing Azure Machine Learning and AWS SageMaker for Hybrid AI Solutions." Microsoft AI Blog, 2023.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details