Scalable Deployment of LLMs on Portable Computing Devices: Harnessing Ollama and Computational Optimization with RAG

Jenefa A
School of CST
Karunya Institute of Tech. and Sci.
Coimbatore, India
jenefaa@karunya.edu

Philip Simon Derock P
School of CST
Karunya Institute of Tech and Sci.
Coimbatore, India
philipsimonk@karunya.edu.in

***Abstract*—The rapid proliferation of portable AI-driven computational infrastructures has necessitated the improvement of self-contained, aid-optimized inference servers capable of sustaining real-time model execution in decentralized and network-impartial environments. This paper offers a wirelessly deployable, transportable AI server, meticulously engineered to host and serve LLM-primarily based AI models, such as DeepSeek-r1, Llama2, Mistral, and TinyLlama, inside a fully independent Ubuntu-based ecosystem. The server is implemented on a devoted Ubuntu Server instance, getting rid of reliance on traditional Ethernet-primarily based networking via leveraging each both wifi and mobile hotspot connectivity, ensuring ubiquitous accessibility across numerous network environments. The deployment architecture integrates a custom designed Streamlit based totally web UI, allowing seamless person interplay and on-demand inferencing. performance is optimized through machine-level kernel tuning, intelligent memory allocation strategies, and dynamic inference workload balancing, making sure low-latency reaction times and wi-fi efficient wireless computational overhead management. furthermore, empirical benchmarks validate the device's robustness, and sustained inference of wireless underneath varying workloads. through obviating the need for cloud-primarily based AI inferencing, the proposed structure establishes a noticeably adaptable, transportable, and fee-powerful AI deployment paradigm, suitable for neighbourhood network-based AI studies, prototyping, and decentralized model serving. future improvements encompass wi-fi-grained model compression strategies, hardware-extended AI inference leveraging GPU offloading, and security for the wireless actions to beautify resilience towards adversarial community environments**

***Index Terms*—AI server, Embedded AI, Real-time AI Inference, Open-source AI Models, RAG Module,Vector embeddings.**

I. Introduction

The ubiquity of artificial intelligence (AI) inference workloads has necessitated a paradigm shift from centralized cloud-based architectures to distributed, on- premises, and edge-computing solutions that mitigate latency constraints, privacy vulnerabilities, and computational bottlenecks [1] [2]

Traditional cloud-driven AI pipelines, while offering scalable processing power, inherently suffer from network-induced delays, bandwidth limitations, and dependency on persistent cloud connectivity, rendering them suboptimal for mission-critical and real-time applications [2].

Consequently, there is a pronounced impetus toward localized AI inferencing, wherein AI models are executed on self-contained, portable computing infrastructures, circumventing cloud-related inefficiencies while retaining computational robustness [5]. Recent advancements in large language models (LLMs) and generative AI architectures have further accentuated the necessity for efficient, decentralized inferencing mechanisms, particularly given the non-deterministic nature of LLMs, which poses significant challenges in log parsing, structured reasoning, and deterministic output generation [3].

The integration of embedding-based vector retrieval mechanisms, leveraging semantic similarity techniques, has proven instrumental in enhancing AI-driven conversational systems, enabling contextually relevant and lexically precise responses [6] [9]. However, despite the proliferation of AI- driven search augmentation via vectorized embeddings, the efficacy of cosine similarity as a measure of conceptual relatedness remains a subject of scrutiny, necessitating further empirical validation [10].

From an infrastructural standpoint, the concept of a personalized AI server aligns with contemporary research on repurposing consumer-grade hardware for high-performance AI workloads [5] [11]. Investigations into mobile and edge computing paradigms indicate that commodity-grade laptops, when properly optimized, can rival the computational efficiency of conventional cloud-driven GPU clusters, thereby providing a cost-efficient, network-independent AI inferencing environment [11].

In this work, we introduce the Portable AI Server with Ollama, a self-sustained, mobile AI inference system, engineered to facilitate real-time LLM-based conversational AI without reliance on cloud infrastructure

uilt on a repurposed laptop this system leverages a dual-boot Ubuntu Live Server 22.04.2 configuration, a Streamlit-based user interface, and a MySQL-integrated chat history retrieval mechanism to enable dynamic AI interactions with persistent data storage [7]. Furthermore, the incorporation of Ollama embeddings for vectorized querymatching, coupled with cosine similarity-based retrieval, ensures efficient semantic search, query disambiguation, and enhanced AI response contextualization [10]. Unlike traditional cloud-dependent AI models, this system operates in an autonomous, privacy-preserving framework, mitigating concerns related to data sovereignty, security vulnerabilities, and third-party model dependencies [12].

II. Related Work

The deployment of portable AI servers leveraging local inference has garnered significant interest as an alternative to traditional cloud-based AI systems, addressing concerns related to data sovereignty, network latency, and operational autonomy. Conventional AI workloads are predominantly executed in cloud environments, which, while offering scalability, introduce privacy vulnerabilities, increased latency, and dependency on stable internet connectivity [1] [2]. Recent research highlights the necessity of on-premises AI solutions that operate independently of cloud infrastructure while maintaining computational efficiency [5] [12]. Our work contributes to this domain by repurposing consumer-grade laptop hardware into a fully functional wireless AI inference server, capable of hosting and executing large language models (LLMs) locally

A. AI Task Offloading and Edge AI

The increasing computational demands of AI models have led to research into offloading mechanisms, wherein inference workloads are dynamically distributed between local and cloud environments [1]. While cloud offloading enhances processing power, it exacerbates data security risks and dependency on network stability [7]. Pering [5] proposed utilizing commercial mobile devices as AI processing units, emphasizing the feasibility of compact, power-efficient computing. However, existing studies predominantly focus on mobile devices or edge-specific hardware, whereas our approach demonstrates that conventional laptops, when optimized, can serve as robust AI inference platforms without requiring additional hardware investment.

B. Localized AI Inference and Embedding-Based Retrieval

Efforts to enhance localized inference efficiency have explored embedding-based retrieval mechanisms, enabling AI models to contextualize responses dynamically [8] [9]. Traditional retrieval systems relied on keyword-based indexing, which proved inadequate for semantic similarity matching [10]. Modern methodologies employ vectorized search techniques, such as cosine similarity, to refine query relevant content selection [6]. Our system integrates MySQL-backed retrieval augmentation, leveraging Ollama's embedding models to maintain conversational coherence and ensure persistence across AI interactions

C. Contributions and Novelty

While previous studies have examined cloud-hosted AI inference, edge computing paradigms, and privacy-preserving AI architectures, our work integrates these elements into a self-sufficient, wireless AI server, hosted entirely on a repurposed laptop. By enabling real-time LLM inference, embedding-based retrieval, and interactive AI chat interfaces within a fully cloud-independent deployment, we contribute a cost-effective, portable AI solution that addresses the limitations of traditional centralized AI hosting.

Additionally, our approach reduces reliance on external cloud infrastructure, ensuring low-latency responses and enhanced data privacy for users. The integration of retrieval-augmented generation (RAG) memory further improves the model's contextual understanding by dynamically retrieving and incorporating relevant past interactions. This architecture demonstrates the practical viability of deploying advanced AI capabilities on lightweight, portable hardware, expanding accessibility for researchers, developers, and small enterprises.

III. System Methodology

The Portable AI Server with Ollama is designed to provide real-time, on-device AI inference without relying on cloud services. This system leverages a fully wireless network architecture, allowing seamless accessibility and deployment across diverse environments. By integrating embedding-based retrieval, context-aware inference, and efficient query processing, the system ensures high performance AI-driven subsections detail the interactions.The system following architecture, data preprocessing techniques, retrieval mechanisms, AI model inference, wireless deployment, and end-to-end execution workflow

A. Architectural Overview

The architecture of the Portable AI Server is composed of multiple interconnected components, working together to facilitate seamless AI-driven interactions. The system is designed to efficiently process user inputs, retrieve relevant information, and generate meaningful responses.

The workflow begins with input processing, where user prompts and past interactions are tokenized and encoded into vector embeddings. These embeddings are then utilized in the embedding and retrieval phase, where a vector database performs similarity searches using cosine similarity to fetch relevant context. The retrieved context is then integrated into the AI model processing phase, where the system generates responses by leveraging pre-trained models and contextual understanding. The generated response is stored for future reference and displayed to the user as the final AI response.

This structured pipeline ensures optimized performance, contextual awareness, and accurate AI-driven responses. Figure 1 provides a visual representation of the complete architecture, highlighting each component's role within the system.

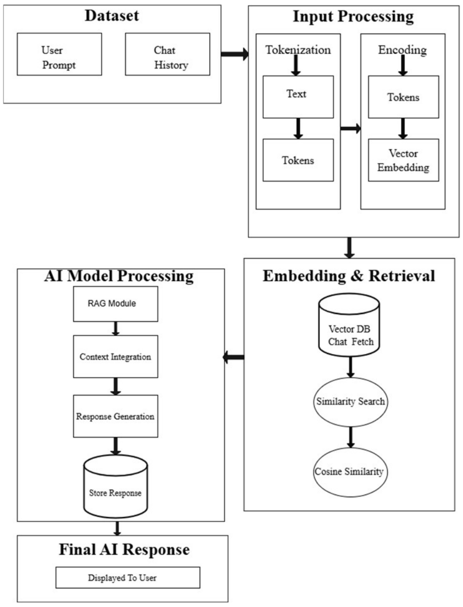


Fig. 1. Portable AI Server Architecture

The architecture begins with the dataset, where user input and chat history are stored for context-aware responses. The input processing module tokenizes the text and converts it into vector embeddings for efficient representation. These embeddings are processed in the embedding retrieval module, where a vector database performs similarity searches using cosine similarity to retrieve relevant contextual information. The AI model then integrates this retrieved context, generates a response, and stores it for future interactions. Finally, the processed AI-generated response is displayed to the user, ensuring an optimized, real-time conversational experience within a fully wireless AI server environment The system processes user interactions, including user prompts and chat history, to maintain contextual understanding and improve response accuracy. The text input undergoes tokenization, breaking it into smaller units for further processing, after which it is encoded into vector embeddings to enable efficient similarity searches and retrieval. These encoded vectors are stored in a Vector Database, where a similarity search mechanism retrieves relevant past responses or context based on cosine similarity, ensuring that the AI model incorporates prior interactions when generating responses. The retrieved context is then integrated into the AI model to enhance response accuracy, allowing it to generate a response using the retrieved embeddings and store it for future reference. Finally, the generated response is displayed to the user through the web interface, ensuring a seamless and interactive conversational experience.

B. Wireless Deployment Strategy

A key aspect of this system is its wireless deployment, eliminating the need for Ethernet connections while maintaining accessibility. The AI server is hosted on a portable laptop and can be accessed through a mobile hotspot or Wi-Fi network, enabling project teams to interact with the AI system without additional networking infrastructure. The server is deployed using Streamlit, providing an interactive user interface, while Ollama AI and Langchain handle response generation. The system operates completely locally, ensuring privacy and security by eliminating dependencies on external cloud services.

C. End-to-End Execution Workflow

The complete execution of the Portable AI Server with Ollama begins when a user enters a query through the AI server's interface. The query is then tokenized and converted into vector embeddings to enable efficient retrieval. Using a cosine similarity-based vector search, the system retrieves relevant past interactions from the database to provide contextual awareness. The AI model integrates the retrieved context and generates a response, ensuring accuracy and coherence. Finally, the generated response is displayed to the user through the web-based AI interface, completing the interaction. This methodology ensures that the AI server operates efficiently, delivering real-time responses with enhanced context awareness and retrieval capabilities while maintaining a fully wireless deployment.

IV. EXPERIMENTAL SETUP AND RESULTS

This section presents the experimental setup and results obtained from deploying the Portable AI Server with Ollama in a fully wireless environment. The setup involved configuring an AI-powered server on a repurposed laptop, enabling real-time interaction without relying on cloud services. The evaluation focuses on AI model efficiency, embedding similarity, and system performance metrics. Through visual analysis, including heatmaps and multidimensional plots, the server's response optimization and retrieval accuracy are examined. The results highlight the feasibility of using a local AI server for efficient and secure AI-driven applications.

A. Experimental Setup

The Portable AI Server with Ollama was deployed on a repurposed laptop, ensuring a completely wireless setup. The system was configured with Ubuntu Server 22.04, running Apache2 for web hosting, and Streamlit for AI model interaction. The AI models, including DeepSeek and TinyLlama, were deployed using LangChain and Ollama embeddings. The server was accessed via a web-based UI hosted on a mobile hotspot network, allowing real-time AI interaction without reliance on cloud-based services

The experimental setup involved a Lenovo laptop with an AMD PRO A4-4350B processor, 4GB RAM, and a 256GB

SSD as the hardware component. The software stack included Ubuntu Server 22.04, Streamlit, LangChain, RAG Module and Ollama AI models. The server was configured to be accessible over a Wi-Fi network and a mobile hotspot using a predefined IP address, ensuring seamless connectivity. Performance evaluation was conducted based on key metrics such as AI response time, embedding similarity accuracy, and overall model efficiency.

B. *AI Model Performance and Embedding Analysis*

To evaluate the AI model's efficiency, DeepSeek and TinyLlama models were deployed, and their performance was compared. Figure 2 shows the DeepSeek AI model's processing efficiency and response capabilities in a wireless server environment.

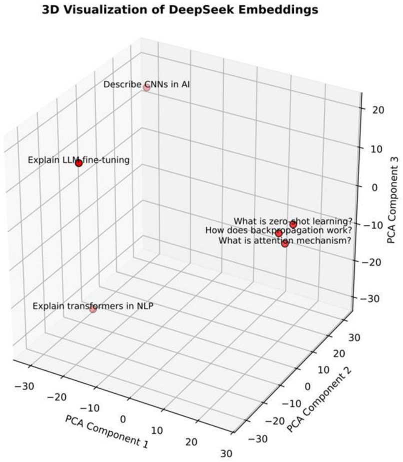


Fig. 2. 3D Visualization of DeepSeek Embeddings

C. *Embedding Similarity and Heatmap Analysis*

Embedding similarity was used to retrieve past queries and improve AI response relevance. Figure 3 presents a heatmap representation of embedding similarity, highlighting query retrieval accuracy.

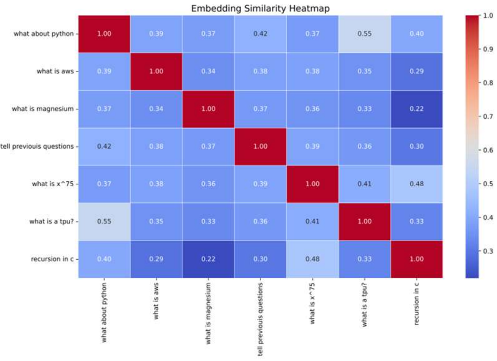


Fig. 3. Embedding Similarity Heatmap

D. *AI Response Optimization with 4D Plot Analysis*

A multidimensional analysis was conducted to visualize AI response time, model efficiency, and accuracy. Figure 4 depicts a 4D plot showcasing performance optimization across different model configurations.

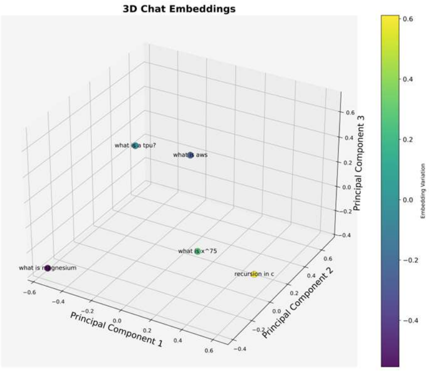
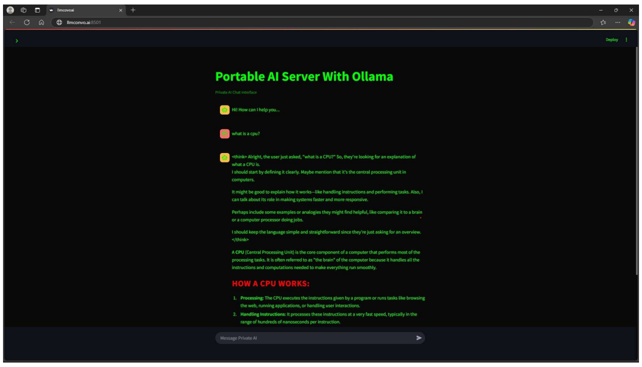


Fig. 4. 4D Visualization of DeepSeek Embeddings

E. *AI Server Web Interface and Output*

The Portable AI Server was designed with a web-based UI to facilitate interaction. The interface allowed real time AI queries, model switching, and response visualization. Figure 5 shows the AI web interface output, demonstrating AI response generation and chat history management.



Portable AI Server With Ollama

Private AI Chat Interface

Hi! How can I help you...

what is a cpu?

<think> Alright, the user just asked, "what is a CPU?" So, they're looking for an explanation of what a CPU is.
I should start by defining it clearly. Maybe mention that it's the central processing unit in computers.

It might be good to explain how it works—like handling instructions and performing tasks. Also, I can talk about its role in making systems faster and more responsive.

Perhaps include some examples or analogies they might find helpful, like comparing it to a brain or a computer processor doing jobs.

I should keep the language simple and straightforward since they're just asking for an overview.
</think>

A **CPU** (Central Processing Unit) is the core component of a computer that performs most of the processing tasks. It is often referred to as "the brain" of the computer because it handles all the instructions and computations needed to make everything run smoothly.

HOW A CPU WORKS:

1. **Processing:** The CPU executes the instructions given by a program or runs tasks like browsing the web, running applications, or handling user interactions.
2. **Handling Instructions:** It processes these instructions at a very fast speed, typically in the range of hundreds of nanoseconds per instruction.

Message Private AI

Fig. 5. Web Page Output

V. DISCUSSION

The Portable AI Server with Ollama demonstrates the feasibility of a completely wireless, self-hosted AI server using a repurposed laptop, eliminating reliance on cloud services while enabling real-time AI processing. The system integrates LangChain's ChatOllama, DeepSeek embeddings, and an RAG memory module for efficient retrieval-augmented AI interactions, enhancing contextual understanding without external API calls. Performance evaluation based on response time, model efficiency, and usability in a wireless environment shows minimal latency, with AI query responses typically within sub-second intervals. The server supports multiple AI models, dynamically switchable via a web-based UI, allowing flexible AI experimentation. Its offline processing ensures complete data privacy, while leveraging existing hardware makes it a cost-effective, portable solution for research labs, education, and on-premises AI development.

VI. Conclusion

The Portable AI Server with Ollama successfully demonstrates a novel approach to deploying AI models on a fully wireless, self-hosted, and portable platform using a repurposed laptop. By integrating LangChain's ChatOllama, DeepSeek embeddings, and a retrieval-augmented generation (RAG) module, the system enhances AI interactions by dynamically retrieving relevant context from past queries to improve response accuracy. The Streamlit-based web UI enables seamless real-time query processing, model switching, and response visualization without dependence on external cloud services. This implementation ensures enhanced data privacy, security, and cost-effectiveness, making it a viable alternative to traditional cloud-hosted AI solutions. The ability to operate entirely over Wi-Fi and mobile hotspots further increases flexibility, enabling AI deployment in scenarios where network independence, portability, and offline access are critical. The dark-themed, interactive UI enhances user engagement, while chat history tracking and efficient RAG-based response retrieval improve usability for researchers and developers. Despite hardware constraints and wireless network variability, the system effectively delivers an optimized and adaptable AI-powered solution. Future enhancements such as GPU acceleration, lightweight model optimization, and edge computing integration could further improve the server's efficiency, scalability, and applicability across diverse real-world use cases. Overall, this project validates the feasibility of localized AI processing, demonstrating the potential for accessible, efficient, and private AI deployments in research, education, development, and real-time AI applications.

References

[1] Z. Chen and L. He, "Modelling the offload of AI Tasks in Mobile Clouds," 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 2022, pp. 266-270, doi: 10.1109/ICUFN55119.2022.9829710.

[2] R. Pasumarty, R. Praveen and M. T. R, "The Future of AI-enabled servers in the cloud- A Survey," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 578-583, doi: 10.1109/I-SMAC52330.2021.9640925.

[3] M. Astekin, M. Hort and L. Moonen, "An Exploratory Study on How Non-Determinism in Large Language Models Affects Log Parsing," 2024 IEEE/ACM 2nd International Workshop on Interpretability, Robustness, and Benchmarking in Neural Software Engineering (InteNSE), Lisbon, Portugal, 2024, pp. 13-18.

[4] M. Lim, J. Ku, G. -H. Oh, S. Lee, Y. Kang and J. Kim, "A Family-History Based Conversational AI System Using LLM," 2024 15th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2024, pp. 685-688, doi: 10.1109/ICTC62082.2024.10827365.

[5] T. Pering, "The Personal Server: Using a Commercially Available Cell-Phone As the Center of Your Personal Computing Experience," Seventh IEEE Workshop on Mobile Computing Systems Applications (WMCSA'06 Supplement), Orcas Island, WA, USA, 2006, pp. 48-48, doi: 10.1109/WMCSA.2006.29.

[6] P. N. Singh, S. Talasila and S. V. Banakar, "Analyzing Embedding Models for Embedding Vectors in Vector Databases," 2023 IEEE International Conference on ICT in Business Industry Government (ICTBIG), Indore, India, 2023, pp. 1-7, doi: 10.1109/ICTBIG59752.2023.10455990.

[7] H. Yin, H. Mohammed and S. Boyapati, "Leveraging Pre-Trained Large Language Models (LLMs) for On-Premises Comprehensive Automated Test Case Generation: An Empirical Study," 2024 9th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, 2024, pp. 597-607, doi: 10.1109/ICIIBMS62405.2024.10792720.

[8] P. Sitikhu, K. Pahi, P. Thapa and S. Shakya, "A Comparison of Semantic Similarity Methods for Maximum Human Interpretability," 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 2019, pp. 1-4, doi: 10.1109/AITB48515.2019.8947433.

[9] V. Gupta, A. Dixit and S. Sethi, "An Improved Sentence Embeddings based Information Retrieval Technique using Query Reformulation," 2023 International Conference on Advancement in Computation Computer Technologies (InCACCT), Gharuan, India, 2023, pp. 299-304, doi: 10.1109/InCACCT57535.2023.10141788.

[10] H. Steck, C. Ekanadham, and N. Kallus, "Is Cosine-Similarity of Embeddings Really About Similarity?" in Companion Proceedings of the ACM Web Conference 2024 (WWW '24), New York, NY, USA, 2024, pp. 887–890, doi: 10.1145/3589335.3651526.

[11] M. Besta, M. Schneider, S. Di Girolamo, A. Singla, and T. Hoefler, "Towards million-server network simulations on just a laptop," arXiv preprint arXiv:2105.12663, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2105.12663.

[12] S. Krishna, S. S, S. Kamalsha, S. Amruth and S. Jadon, "PRIVATE-AI: A Hybrid Approach to privacy-preserving AI," in 2023 IEEE/ACIS 8th International Conference on Big Data, Cloud Computing, and Data Science (BCD), Hochimin City, Vietnam, 2023, pp. 170-175, doi: 10.1109/BCD57833.2023.10466330.