Edge-based Platform for Large Language Models Deployment in Internet of Things Applications

Marc Jayson Baucas, *Member, IEEE,* Petros Spachos, *Senior Member, IEEE*

***Abstract*—Decision-making support systems are a growing trend in Internet of Things (IoT)-based applications. It is due to the introduction of Large Language Model (LLM) deployment platforms. However, cloud-centric approaches struggle with their implementation due to scalability issues. This work proposes an edge-based alternative to improve the scalability of LLM deployment in IoT-based applications. It also presents the incorporation of Raspberry Pis with low-cost, compact, and trending Artificial Intelligence (AI)-based applications, making it an ideal choice as an IoT device for deploying the LLM locally. Experiments evaluate the responsiveness of the LLM under the proposed edge-based platform. The results show its potential in minimizing the likelihood of network scalability issues, while the edge-based approach manages the LLM locally through the Raspberry Pi.**

***Index Terms*—LLM deployment, Edge computing, Raspberry Pi implementation, IoT-based applications**

I. Introduction

LARGE Language Models (LLM) have been a promising complementary piece for many Internet of Things (IoT)-based applications due to their Artificial Intelligence (AI)-trained models. Its contextual training provides timely answers for intelligent decision-making and data analysis [1]. IoT-based applications in industries such as healthcare, automotive, agriculture and many more benefit from LLMs as they can provide decision-making support systems. However, current centralized implementations present a bottleneck to managing all connecting devices accessing the LLM from the cloud server, introducing scalability issues.

On the other hand, an edge-based approach can introduce load balancing by reallocating tasks to capable IoT devices. The rise of more microcomputers with higher processing capabilities, such as the Raspberry Pis [2], shows potential for empowering IoT devices, creating an opportunity for a more edge-based paradigm for IoT-based applications. This work presents the following contributions:

1) An edge-based LLM deployment platform to address this scalability issue within IoT-based applications.
2) A functioning testbed design capitalizing on recent trends in Raspberry Pi implementations in IoT networks.

II. Background

A. IoT-based LLM Deployment Applications

Numerous IoT-based applications deploy LLMs to provide decision-making support to their users [3]. Many industry services use IoT devices to collect and analyze user data. IoT devices in healthcare collect physiological data from patients for health monitoring and early diagnosis. Automotive implementation's IoT sensors gather real-time data on a vehicle's condition for better care. IoT devices in agricultural services collect environmental data to improve farmland productivity and farming conditions. IoT devices enable real-time data collection, providing more opportunities for data analysis. LLMs enhance it by improving user interactability through their context-aware learning [4]. It can provide timely answers, guiding users with more intelligent decision-making.

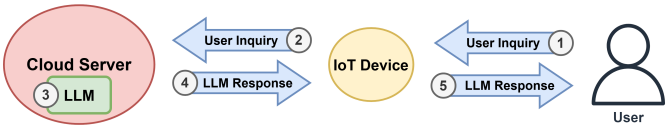
However, one concern with LLMs complementing IoT-based applications is the scalability of the IoT network. The assistive capabilities of LLMs and their context-aware advantages remain a popular topic in industries, leading to many implementations. Most implementations utilize standard centralized cloud servers for their IoT network. As a result, these applications will experience overloading, creating a bottleneck in their operation. Therefore, there is a need for a scalable approach to deploying LLMs within the IoT network.

B. Edge IoT-based approach with Raspberry Pis

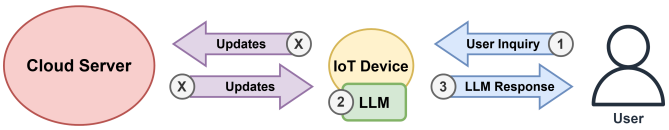
Edge computing brings computation and data storage closer to the data source [5]. Conventionally, most large tasks related to IoT-based applications are within the cloud server. Most centralized network structures create a bottleneck due to the computational load. As a result, complications arise due to the data traffic, preventing IoT applications from functioning effectively. An edge-based approach would unlock IoT devices as an alternative for computational tasks such as LLM inference [6], alleviating the processing load on the server. However, an edge-based approach needs more capable IoT devices to operate effectively.

In IoT-based applications, an emerging technology in implementations is the Raspberry Pi [7]. Raspberry Pis are popular due to their processing capabilities in a compact size. Other advantages include its low cost and modularity, making it ideal for the distributive setup of an edge-based IoT network. This work incorporates the Raspberry Pi 5 to create an edge-based IoT network, offloading the LLM from the cloud to the edge. However, although it can reallocate tasks, it cannot substitute for the cloud server. The Raspberry Pi may have a competent processor, but it is still worse than high-end computers. Therefore, this work examines the advantages of the proposed edge-based platform and evaluates the limitations of this Raspberry Pi and edge-based approach.

Marc Jayson Baucas and Petros Spachos are with the School of Engineering, University of Guelph, Guelph, ON N1G2W1, Canada (e-mail: baucas@uoguelph.ca; petros@uoguelph.ca).



(a) Cloud-centric configuration.



(b) Edge-based configuration.

Fig. 1: Configurations used for testing the proposed platform.

III. Platform Overview and Design

A. Overview

Previous work in [8] carries a similar approach to the proposed edge-based platform. However, the current work emphasizes the combination of an edge-based approach and the Raspberry Pi. It also presents its effectiveness in deploying LLMs locally and a platform with a cloud-edge IoT network configuration. It has a central cloud server for managing more general and higher-level tasks, while the edge IoT devices handle the other smaller computational tasks. Recent works and implementations on LLM deployment, such as [1], [9], [10], heavily emphasize server-side management through the cloud or local servers. However, this work examines the effectiveness of managing LLMs through edge IoT devices instead. It aims to verify the feasibility of an edge-based approach for LLM deployment platforms for IoT applications. It also provides a testbed design using Raspberry Pis.

B. Components and Design

1) General Specifications: The platform runs code on Python 3. It has two components: the cloud server and the edge IoT device. They communicate wirelessly via socket communication on a shared Wi-Fi network. There are two configurations for testing the platform: cloud-centric and edge-based. A diagram comparing these two configurations and their flow of operations is in Fig. 1. The cloud-centric setup, as shown in Fig. 1a, focuses all computational tasks on the cloud server. The IoT device will take the user's inquiry and send it to the cloud server. Next, the server will enter the prompt in the LLM. Afterwards, it sends the LLM's response back to the IoT device. The edge-based approach, as shown in Fig. 1b, reallocates the LLM to the IoT device, allowing it to process the user inquiry locally. Its communication with the server will only be for updates on IoT device code from the server or notifications of errors or operational issues with the edge IoT device. These updates are asynchronous with the communication between the edge IoT device and the user.

2) Cloud Server Setup: The platform uses a commercial computer as its cloud server. It is an MSI Katana laptop running a Windows Linux Subsystem. Other specifications of this computer are an Intel Core i7 processor and 32 GB of RAM. It runs a Python script that initializes a socket server. This server continuously listens on an open port for clients requesting access. A commercial computer is selected instead of a third-party cloud server service because it allows testing that eliminates extraneous variables, such as additional latency. This latency is because of distance and compatibility issues with the third party's cloud assets.

3) Edge IoT Device Setup: The platform uses a Raspberry Pi 5 as its edge IoT device. It runs on a 64-bit Raspbian OS. Other specifications of this device are 8 GB RAM and an ARM Cortex-A76 processor. It runs a Python script that initializes a socket client. This client continuously communicates with a server's listening port. Raspberry Pis are selected because they are low-cost and modular, enabling rapid prototyping. Also, its continuous technological improvements and wide-ranging open-source support make it more flexible for development and a seamless fit for IoT-based applications.

4) LLM Setup: The LLM component of the platform uses a combination of Ollama and LangChain as Python libraries. Ollama enables the device to install and run LLMs locally. It pulls the specified model from an online repository and deploys it through a standalone server. LangChain is a framework for interfacing with the LLM through a Python script. It allows the device to enter input prompts and collect inferences from the LLM. Ollama and LangChain are ideal options because they are compatible with the Raspberry Pi's Linux-based OS and hardware constraints as open-source Python libraries. Both are also well-documented with an active and collaborative community. It promotes up-to-date and sustainable development. Ollama has access to a vast selection of open-source LLMs through its online repository and other sources such as Hugging Face.

This platform implements a testbed to evaluate the performance of the Raspberry Pi in running LLMs locally. It uses models from Hugging Face, a leading source for downloading LLMs for research and development. It has a list called the "Open LLM Leaderboard" [11], ranking all models based on accuracy scores from different evaluation metrics compiled by an evaluation framework [12]. This work uses the scoring from the Instruction-Following Evaluation. It focuses on the model's ability to follow instructions and formatting. Concurrently, it generates answers that must be contextually close to the question [13]. This selection matches the nature of most LLM-based IoT applications. It focuses on strict attention to inquiry format and context to provide the best guidance as a decision-making support system. It only chooses from the top LLMs listed by Hugging Face that can run on edge IoT devices and have official providers. Since the Raspberry Pi 5 has 8 GB of RAM and a low-end processor, it can only handle models with smaller sizes. As a result, it can only run compact models built for lightweight devices with low processing capabilities. Other qualities considered when selecting LLMs included community support, up-to-date versions, and whether the software is open-source. The models that did not meet this work's criteria were not selected. The top three LLMs are IBM's Granite 3.2:2B [14], Meta's Llama 3.2:1B [15], and Google's Gemma 2.0:2B [16].

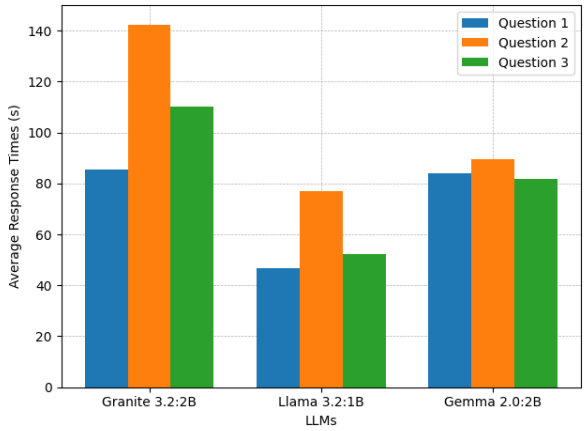


Fig. 2: Average response times of LLMs deployed by the Raspberry Pi 5 under the edge-based configuration.

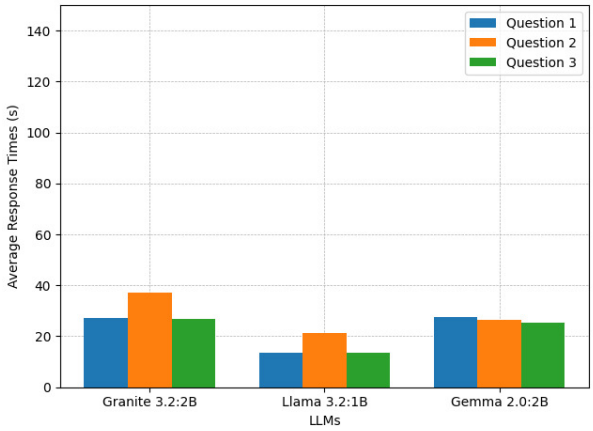


Fig. 3: Average response times of LLMs deployed by the commercial computer under the cloud-centric configuration.

IV. Evaluations and Results

A. Raspberry Pi Performance Evaluation

The testbed aims to investigate the feasibility of using a Raspberry Pi as the edge IoT device of the platform. It compares the platform's ability to run the three LLMs from Hugging Face. It evaluates each LLM with three questions selected arbitrarily. Each question represents three different trending decision-making support systems: healthcare, farming, and automotive. This choice tests the testbed's functionality with varying questions. The intention was not to claim a complete generalization to all types of questions. Instead, future works will consider adding more questions for better coverage. The questions this testbed iteration uses are:

1) Define Body Mass Index (BMI) and explain how to calculate it.
2) Explain drip irrigation and why farmers use it in modern agriculture.
3) What is cruise control, and how does it help during long trips?

The testbed will send each question ten times and average the measured response times. It is the time it takes for the client to get a response from the LLM after sending its inquiry. It varies depending on the configuration. For the cloud-centric, the time includes the communication time between the client and the server. For edge-based, there is no communication between the client and server. Therefore, it only measures the processing time.

First, the test evaluates the Raspberry Pi 5. The average response times using the Raspberry Pi to deploy them are shown in Fig. 2. The difference in average times per question shows how each LLM performs differently. Granite showed more volatility across its questions, suggesting that the type of question can impact an LLM's inference speed. Gemma supports this hypothesis as it shows stability with its relatively close average response times. Llama yielded the lowest times, suggesting that it is the most optimal for the Raspberry Pi to use. At the same time, the tests had no issues using each LLM. This result shows the feasibility of using the Raspberry Pi 5.

Next, the test evaluates the cloud-centric approach, where the cloud server deploys the LLM. The LLM's response times using the commercial computer to deploy them are shown in Fig. 3. It presents the same trends as the Raspberry Pi's results. However, when the commercial computer deploys the LLM as the cloud server, its average response times are within a range of 2-3 times faster. This difference suggests that although the Raspberry Pi 5 can run the LLMs, the disparity in average response times shows a discrepancy. This difference is due to the Raspberry Pi's technological limitations, such as a lower-end processor and smaller memory capacity.

This test only reveals the limitations of the Raspberry Pi 5 when compared directly to a higher-end processor. The proposed platform combines Raspberry Pis with edge computing, maximizing its strengths to create a more viable approach for LLM deployment in IoT-based applications. The advantage of an edge-based approach is in its distributed and load-balancing structure. Therefore, the next test will simulate a growing network. It will measure the average response time when deploying the LLM with each configuration.

B. Edge-based vs Cloud-centric Comparison

This test compares the two configurations, considering the limitations of the Raspberry Pi 5 when deploying the LLM. It will measure the platform's responsiveness as it receives an inquiry under the two types of configurations. The test will gradually increase the number of IoT devices connected to the server from 1 to 2, 4, 6, 8, and 10. With the cloud-centric configuration, the IoT device sends the inquiry to the server and awaits its response. With the edge-based configuration, the IoT device processes the inquiry locally and presents it to the user. This test will measure the average response time 10 times for each increase in network size. Also, it uses only one LLM and a question to ensure a consistent measurement. Therefore, it uses Llama because it performed the best in the previous test and question 3, which was arbitrary.

The responsiveness of the two configurations is shown in Fig. 4. The results reveal that as the network grew, the average response time of the cloud-based configuration increased.

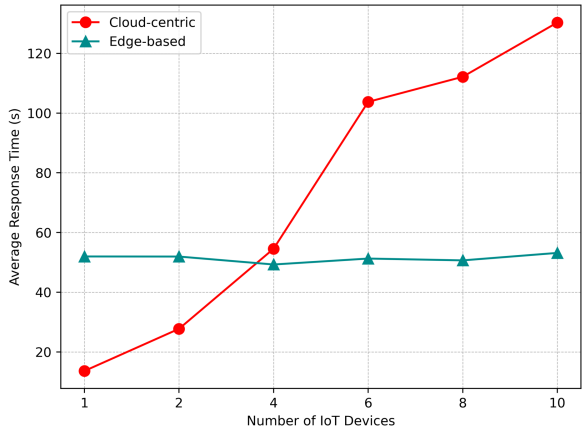


Fig. 4: Average response time of cloud-centric and edge-based configurations as IoT devices increased.

Meanwhile, the edge-based configuration remained relatively consistent. This trend suggests the scalability of the edge-based approach. The cloud-centric implementation eventually ran into a bottleneck, limiting its responsiveness. The point where the two plots intersect indicates when the cloud-centric configuration is now less responsive than the edge-based configuration. Another observation is that the Raspberry Pi 5 was enough to create a more responsive platform for deploying LLMs. Its ability to locally deploy the LLM allows the IoT network to operate under an edge-based configuration.

The combination of Raspberry Pis and edge computing is a feasible platform. The results showcase its effectiveness in deploying LLMs in IoT-based applications. Both benefit one another, creating a combination that promotes a more responsive platform. The Raspberry Pi showed limitations in computational capabilities compared to higher-end computers. However, the edge-based approach's advantages surpass these limitations by addressing the latency issues of a cloud-centric configuration. An edge-based approach needs a capable IoT device for reallocating processes from the cloud, balancing the computational load across the IoT network. The results of the previous test show that the Raspberry Pi 5 can deploy LLMs. Also, its modular, low-cost, and compact design makes it ideal for an edge-based approach to IoT-based applications. Overall, these tests show the feasibility of this combination for deploying LLMs in IoT-based applications.

C. Future Works and Discussion

Currently, the questions only touch on healthcare, agriculture, and automobile support systems. It does not generalize to all types of questions outside of these types of support systems. Future iterations will improve the scope of the testbed by adding questions highlighted in other relevant works. Other improvements aim to compare the design with existing implementations. Another is to incorporate higher-end LLMs, highlighting the processing advantages of a cloud-centric configuration and compare it against using an edge-based IoT network. Lastly, breaking down the response time into multiple components creates a more systematic analysis. These additions to the testbed design reinforce the discussion of the trade-offs between accuracy and response time.

V. Conclusion

This work proposes an edge-based approach to deploying LLMs in IoT-based applications using Raspberry Pis. It reallocates the processing load concentrated in the cloud to IoT devices, reducing the likelihood of this bottleneck. Raspberry Pis are low-cost, modular, and compact, making them excellent options for IoT-based applications. This work examined the capabilities of the Raspberry Pi to deploy LLMs to answer user inquiries. The results present its viability as an edge IoT device, but a comparison with the performance of a higher-end computer exposes its limitations. Another test compares the scalability of the edge-based and cloud-centric configuration. As the network grew, the cloud-centric configuration's response time increased, revealing scalability issues. The edge-based configuration remained consistent even with the Raspberry Pi's limitations. Therefore, combining the Raspberry Pi and the edge-based approach presents a feasible platform for deploying LLMs in IoT-based applications. The Raspberry Pi provides a capable IoT device, while the edge-based approach compensates for its processor limitations.

References

[1] B. Xiao *et al.*, "Efficient prompting for llm-based generative internet of things," *IEEE Internet of Things Journal*, vol. 12, no. 1, pp. 778–791, 2025.

[2] A. Jevremovic *et al.*, "Energy efficiency of kernel and user space level vpn solutions in aiot networks," *IEEE Open Journal of the Computer Society*, vol. 6, pp. 199–210, 2025.

[3] O. Friha *et al.*, "Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 5799–5856, 2024.

[4] A. Neyem *et al.*, "Toward an ai knowledge assistant for context-aware learning experiences in software capstone project development," *IEEE Transactions on Learning Technologies*, vol. 17, pp. 1599–1614, 2024.

[5] M. J. Baucas and P. Spachos, "Edge-based data sensing and processing platform for urban noise classification," *IEEE Sensors Letters*, vol. 8, no. 5, pp. 1–4, 2024.

[6] M. Zhang *et al.*, "Edgeshard: Efficient llm inference via collaborative edge computing," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 13 119–13 131, 2025.

[7] S. Karthikeyan *et al.*, "A systematic analysis on raspberry pi prototyping: Uses, challenges, benefits, and drawbacks," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 397–14 417, 2023.

[8] M. J. Baucas *et al.*, "Private blockchain-based edge iot platform for secure large language model services," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, 2025, pp. 1–6.

[9] J. Zheng *et al.*, "Intent-based multi-cloud storage management powered by a fine-tuned large language model," *IEEE Access*, vol. 13, pp. 72 736–72 753, 2025.

[10] D. F. Pedroso *et al.*, "Anomaly detection and root cause analysis in cloud-native environments using large language models and bayesian networks," *IEEE Access*, vol. 13, pp. 77 550–77 564, 2025.

[11] C. Fourrier *et al.*, "Open llm leaderboard v2," https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

[12] L. Gao *et al.*, "A framework for few-shot language model evaluation," Sep. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5371628

[13] J. Zhou *et al.*, "Instruction-following evaluation for large language models," 2023. [Online]. Available: https://arxiv.org/abs/2311.07911

[14] G. Team and IBM, "Granite-3.2:2b," https://ollama.com/library/granite3.2:2b, February 2025, long-context AI model fine-tuned for thinking capabilities.

[15] M. AI, "Llama 3.2:1b," https://ollama.com/library/llama3.2:1b, September 2024, accessed: 2025-05-13.

[16] G. Team, "Gemma," 2024. [Online]. Available: https://www.kaggle.com/m/3301