Generative Inference of Large Language Models in Edge Computing: An Energy Efficient Approach

Xingyu Yuan, *Student Member, IEEE*, He Li, *Member, IEEE*, Kaoru Ota, *Member, IEEE*,
Mianxiong Dong, *Member, IEEE*
Department of Sciences and Informatics
Muroran Institute of Technology
Muroran, Hokkaido, Japan
Email: 23096502@muroran-it.ac.jp, heli@mmm.muroran-it.ac.jp, {ota, mx.dong}@csse.muroran-it.ac.jp

2024 International Wireless Communications and Mobile Computing (IWCMC) | 979-8-3503-6126-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/IWCMC61514.2024.10592339

***Abstract*—Large Language Models (LLMs) have demonstrated remarkable proficiency in generating text and producing fluent, succinct, and precise linguistic expressions. Limited battery life and computing power make it challenging to process LLM inference tasks in mobile devices. Intelligent edge computing brings the opportunity to help users process LLM inference tasks in real-time by offloading computations to nearby edge devices. However, due to the undetermined relationship between various task requirements and offloading configurations, inefficient offloading leads to unaffordable additional energy consumption, especially for intelligent tasks. This paper first investigates the energy consumption issue with different offloading configurations and task requirements in an intelligent edge testbed. According to the preliminary experiment results, we formulate the LLM offloading problem as a multi-armed bandit (MAB) problem and then use an upper confidence bound (UCB) bandit algorithm to find the energy-efficient offloading configurations. Extensive simulation results show that our approach enhanced the energy efficiency for offloading LLM inference tasks with different requirements in the intelligent edge environment.**

***Index Terms*—Large Language Models (LLMs), Energy Efficiency, Intelligent Edge Computing**

I. Introduction

Large Language Models (LLMs) such as OpenAI's GPT series and its derivatives have shown powerful capacity in various Natural Language Processing (NLP) tasks across multiple languages [1]. These models consume significant computational and storage resources for both training and inference. Therefore, given the constraints of devices and environmental resources, most LLM applications are typically hosted on high-end cloud servers [2].

Intelligent edge computing facilitates real-time inference on local devices by placing computing servers close to the data source. This strategy significantly reduces data transmission latency to the cloud and provides better privacy [3]. However, unlike ordinary edge AI tasks, the LLM inference task requires more computational resources and storage support. Given edge devices' typically limited battery life, minimizing the energy consumption of task inference is critical to ensure edge devices' stable and reliable operation.

Research relevant to edge computing focuses on optimizing the location of task offloading to improve the energy efficiency of the task process [4], [5]. Usually, these methods focus on selecting only one configuration for task offloading, such as a different offloading device. For the LLM inference task, the energy consumption of the inference is affected by various factors, such as the number of parameters for LLM, the length of the input text, and so on. Therefore, optimizing the selection of multiple offload configurations will be a better option to improve energy efficiency. On the other hand, meeting the task requirement of the LLM inference task is also essential. In contrast, task requirements can be inference accuracy, inference time, or a combination. Notably, task requirements from the user's perspective are intricately intertwined with energy consumption requirements from the server's perspective. For example, using a low-power device or an LLM with fewer parameters may reduce energy consumption. Still, it will increase the inference time of the task or achieve low accuracy. So, it is challenging to meet the task requirements of each user while optimizing the selection of multiple offload configurations.

We are conducting preliminary experiments within an edge computing testbed to address this issue and identify offloading configurations critical for task requirements and energy consumption. Modifying these configurations allows us to observe the corresponding shifts in energy consumption and task requirements. The experimental results, detailed in Section III, show that a suitable selection of configurations can significantly improve energy efficiency and ensure task requirements. Nevertheless, the automatic assignment of suitable configurations for offloading LLM inference tasks in edge environments is a complex challenge. Unlike general computing tasks at the edge, LLM inference tasks have multiple configurations, and the combination of these configurations has unpredictable effects on energy consumption and task requirements, making configuration selection more complex.

This paper presents a new approach to make energy use more efficient when offloading large language model (LLM) inference tasks to the edge environment. Our method starts by modeling this challenge as a multi-armed bandit (MAB) problem. This helps us to understand how different config-

979-8-3503-6126-1/24/$31.00 ©2024 IEEE

urations affect the energy consumption of Intelligence edge computing when processing LLM tasks. To solve this MAB problem, we use the Upper Confidence Bound (UCB) algorithm. In asymptotic settings, UCB is shown to achieve close to optimal regret limits, making it attractive for multiarm and large-horizon scenarios. We'll show how effective this is in Section V. The main contributions of this paper are as follows:

1) Firstly, we perform preliminary experiments, and the experimental results show the impact of different configurations on the task requirements and energy consumption of the LLM inference task in an edge computing environment.
2) Next, we define an MAB issue to determine the best configurations for offloading LLM tasks in an intelligent edge environment. We use the UCB algorithm to solve the MAB problem, effectively determining the best configurations while considering different task requirements.
3) Finally, we compare the UCB algorithm with two standard algorithms. Numerous simulations have shown the superior performance of the UCB algorithm over other algorithms, highlighting its advantage in our energy efficiency problem.

The rest of this paper is organized as follows. We discuss the related work on LLM and intelligent edge computing in Section II. Section III presents the pre-experimental setting and shows experimental results. The MAB problem formulation is described in Section IV. Performance evaluation is discussed in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Edge computing has been an essential area of research in computer networking for over a decade. More recently, there has been an increasing focus on energy-efficient computing, particularly on edge environments. In this section, we review some related work that explores these aspects.

In addressing the complexities of energy optimization in edge computing, recent literature presents a multi-faceted approach to improving efficiency within the Internet of Things (IoT) and Cyber-Physical Systems (CPS). Na Su et al. [6] address underutilized computing resources in wired edge devices within the Industrial Internet of Things (IIoT), proposing computational task assignment and resource allocation algorithms to balance energy consumption with latency requirements. Their methodology paves the way for improved computational performance in delay-sensitive networks. Focusing on the burgeoning needs of 5G and emerging 6G networks, Abegaz Mohammed Seid et al. [7] present a deep federated learning-based framework for airborne mobile edge computing (MEC) servers in smart city CPS. Their hierarchical model significantly improves the management of ultra-low latency applications, addressing the expected congestion of terrestrial MEC servers. In the context of smart buildings, Muhammad Ibrar et al. [8] developed REED, a model that integrates software-defined networking, edge computing, and device-to-device (D2D) communication. This model aims to minimise energy consumption and latency, using deep deterministic policy gradient algorithms to effectively manage the high density of IoT devices.

In recent years, with the rapid development of artificial intelligence, more and more AI tasks are being deployed on edge devices. First, Sha Zhu et al. [9] investigate the energy requirements of AI-driven applications within the IIoT and propose an intelligent edge computing framework with a novel scheduling algorithm. Their approach demonstrates a significant reduction in energy consumption compared to traditional methods. Next, Galanopoulos et al. [10] address video analytics in wireless services, a domain that requires extensive data processing. They propose an Automated Machine Learning (AutoML) framework for dynamically configuring service and network parameters to improve the accuracy of the analysis while maintaining frame rate constraints. Their Bayesian online learning algorithm demonstrates adaptability and effectiveness, optimizing real-time configurations. Yuan et al. [11] address the challenge of semantic segmentation in computer vision, which is hampered by user devices' limited computing power and battery life. They present an approach to improve energy efficiency by offloading computations to neighboring devices, formulating this as a constrained multi-armed bandit problem. Their improved upper confidence bound algorithm significantly increases the energy efficiency of offloading these intensive tasks in edge environments. Finally, in healthcare, S. Abirami et al. [12] explore the potential of wearable IoT (wIoT) devices in remote health monitoring systems for diabetic patients with cardiovascular disease. They proposed a health support system named EESE-HSS to improve energy efficiency in the smart cloud edge paradigm. This system focuses on rapid emergency diagnosis with an energy-efficient edge intelligence framework, highlighting the importance of reduced latency and improved energy efficiency in critical healthcare applications.

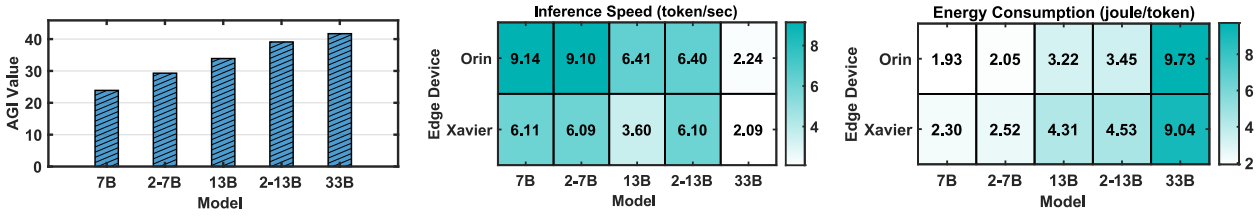
Overall, these works significantly contribute to the energy-efficient edge computing field. However, we did not find any relevant research on the energy efficiency of large language models in edge computing, and our work fills this gap.

III. Preliminary Experiment

In this section, we present the details of our preliminary experiments.

Given the high computational demands of large language model inference tasks, we selected the NVIDIA Jetson AGX Orin and NVIDIA Jetson AGX Xavier for our experiments. These platforms are renowned for their high-performance capabilities, particularly in AI and robotics applications. In the edge environment, we run text-generation-webui in Docker. The text-generation-webui is a radio-based interface for running large language models such as LLaMA, GPT-J, and OPT. For the Large Language Model, we choose five models trained by Meta, LLaMA-7B, LLaMA-13B, LLaMA-33B, LLaMA-2-7B, and LLaMA-2-13B [13], which is deployed to two devices. We are using GPTQ-for-LLaMa as a model loader.

To ensure maximum device performance, we activated the Jetson clocks and monitored the power consumption of each



(a) AGI Value (b) Inference Speed (c) Energy Consumption

Fig. 1. AGI Value, Inference Speed, and Energy Consumption for Different Configuration

device in real time using the Jetson Power GUI. Before starting the experiment, we defined the accuracy of the inference task. We use the AGI Eval [14] provided by Meta to evaluate the model's accuracy. The AGI Eval is a widely acknowledged benchmark, renowned for its comprehensive evaluation of underlying models in the context of human-centered, standardized assessments such as advanced placement tests., medical licensing examinations, mathematics competitions, and bar exams. The AGI value, as depicted in Fig. 1a, represents the quantitative score derived from the AGI Eval tool. This score quantifies the model's effectiveness and its capability to mimic human-like reasoning and decision-making across the aforementioned standardized tests. A higher AGI value indicates superior performance, suggesting that the model demonstrates a closer approximation to human-level competencies in the evaluated domains.

On the other hand, the inference speed is also a critical evaluation standard, so we evaluated the inference speed of the five models on the device, and the results are shown in Fig. 1b. The fastest inference configuration was LLaMA-7b on Jetson AGX Orin at 9.14 tokens per second, and the slowest inference configuration was LLaMA-33b on Jetson AGX Xavier at 2.09 tokens per second Combining with the AGI Eval value, we can observe that LLaMA-7B and LLaMA-2-7B produce similar speeds when performing inference, but LLaMA-2-7B has a higher AGI Eval value. The same is true for LLaMA-13B and LLaMA-2-13B. As for LLaMA-33B, it has the highest AGI Eval value but has a slow inference speed.

Finally, we evaluated the energy consumption of the five models for inference on two edge devices. Fig. 1c shows the energy consumption required to generate one token for different configuration values. We can see running the LLaMA-33B model on the Jetson AGX Orin platform results in energy consumption of approximately 9.73 joules per token generated. In contrast, operating the LLaMA-7B model on the same platform incurs a significantly lower energy expenditure, estimated at 1.93 joules per token. This indicates that the energy consumption associated with the LLaMA-33B on the Jetson AGX Orin is nearly 5x greater than that of the LLaMA-7B for analogous tasks, disregarding specific accuracy and inference speed requirements. However, our analysis revealed a noteworthy finding when task requirements were factored in. For instances where the accuracy threshold exceeds 33.9, and the desired inference speed surpasses 3.6 tokens per second, the most energy-efficient configuration does not involve the LLaMA-7B model on the AGX Xavier platform, as might be expected. Instead, the LLaMA-7B model operating on the Jetson AGX Orin emerges as the optimal choice regarding energy efficiency.

The experimental results show that the different configurations have a complex impact on task energy consumption and task requirements when inference tasks of the large language models on edge devices. The relationships between these configurations show unpredictable correlations. This makes determining the optimum configuration value challenging before processing the task.

IV. System Model

This section provides a comprehensive overview of our system framework and explains the modeling of the system's energy consumption.

A. System model

We proposed a system that uses intelligence edge computing to make generative inference tasks more energy-efficient, as shown in Fig. 2. In our setup, users upload tasks to the edge server and choose the lowest acceptable inference speed and AGI value. We denote the number of tasks as $n = 1, 2, \ldots, N$ and denote minimum acceptable inference speed as δ_n, minimum acceptable AGI value as λ_n. The accuracy and inference speed represent two pre-defined sets of reference values we offer. These benchmarks are obtained from experiments conducted on various devices and models.

Our system runs in discrete rounds, denoted t, where $t = 1, 2, \ldots, T$, each round being assigned a specific configuration value. As highlighted in Section III, two critical configurations (edge device and the number of model parameters) significantly affect a task's energy consumption, inference speed, and accuracy. However, the interplay between these configurations is complex and not easily predictable. Our goal is to minimize the energy consumption of tasks while satisfying the task's inference speed and accuracy constraints of the task by choosing the suitable configuration value.

Therefore, we select the type of edge device from a finite set $\mathcal{D}$ and choose the number of model parameters from a finite set $\mathcal{P}$. So we define the configuration value on round t as:

$$z_t = (d_t, p_t) \in Z \equiv \mathcal{D} \times \mathcal{P}. \tag{1}$$

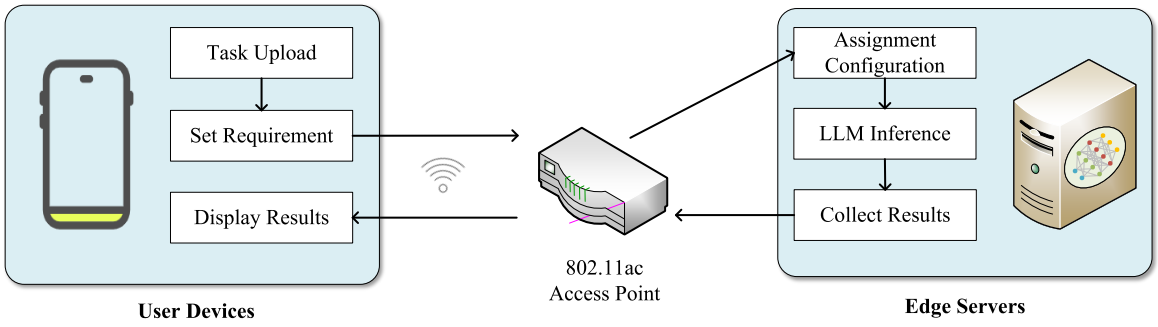


Fig. 2. System Framework

B. System Energy Consumption

In edge servers, the primary source of energy consumption is the generative inference of tasks. After selecting the configuration value of the task, the edge servers adjust this configuration and start the generative inference. When a task is completed, the generated text is sent back to the user via a wireless network. The edge server records the energy consumed during the generative inference process. To represent the inference energy consumption, we use the notation $p_t^{inf}(z_t)$, where z_t denotes the configuration value chosen for round t. We then aggregate the energy consumption in each round to formulate a measure of the total energy consumption by the system during the generative inference phase:

$$p_{total}^{com} = \sum_{t=1}^{T} p_t^{com}(z_t). \tag{2}$$

In our system, the energy consumption for data transmission is calculated as the product of the transmission power and the transmission time. We denote the transmission time in round t as l_t. Transmission time depends on the size of the data to be transmitted, represented as d_t, and the transmission rate, R^{tran}. The formula for calculating the transmission time is, therefore, $l_t = d_t/R^{tran}$. The transmission rate, R^{tran}, is derived based on the principles of the Shannon-Hartley theorem, formulated as $R^{tran} = B\log_2(1 + S^{tran}/N^{tran})$, where B is the channel bandwidth, S^{tran} is the average power of the received signal, and N^{tran} is the power of the transmitted noise. So, the total energy consumption by the system during the transmission phase is:

$$p_{total}^{tran} = \sum_{t=1}^{T} p_t^{tran}, \tag{3}$$

where $p_t^{tran} = o^{tran} l_t$, and o^{tran} is the transmit power when transmitting tasks.

In summary, we can define the energy consumption of the system by adding the energy consumption of generative inference and the energy consumption of transmission together:

$$P = p_{total}^{tran} + p_{total}^{inf}. \tag{4}$$

C. Problem Formulation and Solution

Inference speed and accuracy are two critical metrics for evaluating our system's performance of LLM inference tasks. We denoted the inference speed when configuration value z_t is applied to task n by $S_n(z_t)$. And denoted the accuracy when configuration value z_t is applied to task n by $A_n(z_t)$.

$$\begin{aligned} S_n(z_t) - \delta_n > 0 \\ A_n(z_t) - \lambda_n > 0. \end{aligned} \tag{5}$$

Our goal is to maximize P^{-1} while maintaining task requirements.

$$\begin{aligned} \text{Maximize} \quad & P^{-1} \\ \text{subject} \quad & S_n(z_t) - \delta_n > 0 \\ & A_n(z_t) - \lambda_n > 0 \\ & P \neq 0. \end{aligned} \tag{6}$$

The UCB algorithm is a popular solution to the MAB problem. Unlike approaches that rely on random selection for exploration, the UCB algorithm dynamically adjusts its exploration-exploitation balance based on evolving observation for the operational environment. This adaptive mechanism allows the algorithm to progressively explore and refine configuration values expected to yield the highest returns.

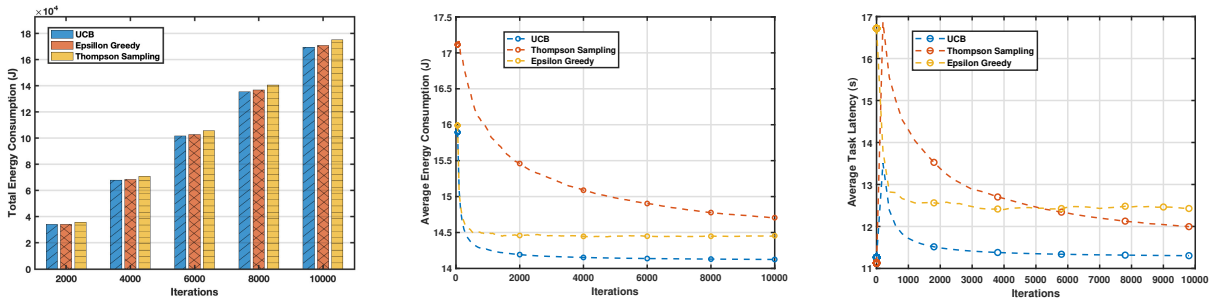
In UCB algorithm, the configuration value z chosen at time step t, is given by:

$$z_t = \arg\max_{z \in Z} \left[V_t(z) + c\sqrt{\log t / N_t(z)} \right], \tag{7}$$

where $V_t(z)$ denotes the estimated value of the configuration z at round t. The term c represents the confidence level, calibrating the algorithm's degree of exploration. Additionally, $N_t(z)$ signifies the count of instances where the configuration value z has been selected before round t.

V. Performance Evaluation

For our simulation framework, we set the average signal-to-noise ratio (SNR) per user to 50 dB and the channel bandwidth to $B = 30$ MHz. In addition, we adopt the IEEE 802.11ac standard for the wireless channel model. Then, as comparison algorithms, we choose the Thompson Sampling algorithm and the Epsilon Greedy algorithms, two other commonly used algorithms for the MAB problem.



(a) Total Energy Consumption (b) Average Energy Consumption (c) Average Task Latency

Fig. 3. Comparison of Energy Consumption and Latency

Algorithm 1 UCB for LLM Inference Task Offloading

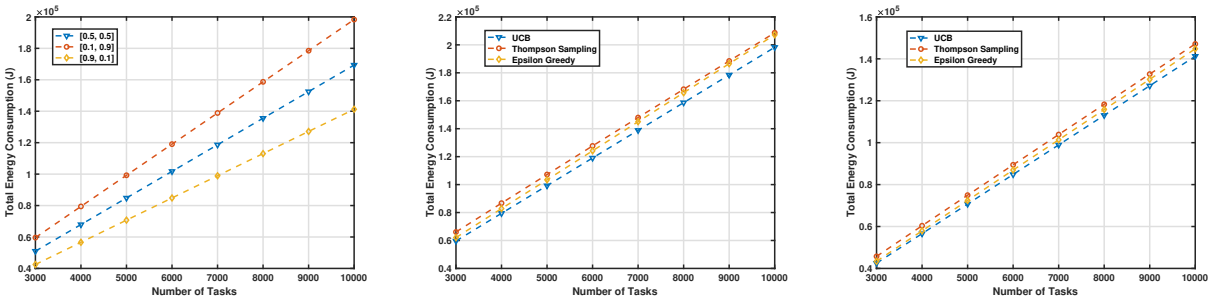
Initialization:
  Initialize δ_n, λ_n, $V = 0$, $c = 1$.
Input:
  Configuration set Z,
  Confidence value c,
  Inference speed threshold δ_n,
  Accuracy threshold λ_n,
for $t = 1, \ldots$ **do**
  $z_t = \arg\max_{z \subseteq Z} \left[V_t(z) + c\sqrt{\log t / N_t(z)} \right]$
  get round reward z_t
  get round inference speed value $S_n(z_t)$
  get round accuracy value $A_n(z_t)$
  total reward $V \leftarrow V + z_t$
  if $S_n(z_t) - \delta_n < 0$ **or** $A_n(z_t) - \lambda_n < 0$ **then**
    go back to step 6 to reselect z_t
  end if
end for

First, we set the accuracy requirement as $\lambda_n > 20$ and the inference speed requirement as $\delta_n > 2$, and we evaluate the total energy consumption of the three algorithms in 7,000 iterations. The comparative results in terms of total energy consumption are shown in Fig. 3a. It can be seen that Thompson Sampling has a higher energy consumption compared to the other algorithms. From 0 to 1,000 iterations, the performance of the Epsilon Greedy and UCB algorithms are similar, averaging around 16,000 joules. However, as the number of iterations increases (from 1,000 to 7,000), the superiority of the UCB algorithm becomes clearer. By the 7,000th iteration, the total energy consumption of the Epsilon Greedy algorithm reaches about 110,400 Joule, while the UCB algorithm remains at a lower level of about 99,200 Joule. This trend indicates that the UCB algorithm is more effective than the Epsilon Greedy algorithm in optimising energy efficiency in MAB problems.

Then, we evaluate the average energy consumption of three algorithms in 10,000 iterations. The results are shown in Fig. 3b. The average energy consumption of all three algorithms decreases as the number of iterations increases, but the UCB algorithm is the first to converge. We note that the UCB algorithm converges after 1,000 iterations. After 10,000 iterations, the average energy consumption of the UCB algorithm is 14.2 joules, compared to 14.5 joules and 14.7 joules for the Epsilon Greedy and Thompson Sampling algorithms, respectively. Fig. 3c shows the average task latency of three algorithms, and we can see that the UCB algorithm can always maintain the lowest average latency.

Considering the tasks have different requirements, we added three sets with different requirements for comparison. We define the original requirement as [0.5, 0.5], which means that half of the tasks are accuracy sensitive ($\lambda_n > 40$), and the other half of the users are inference speed sensitive ($\delta_n > 9$)). In addition, we chose two unique combinations of requirement [0.1, 0.9] and [0.9, 0.1], The former implies that 90% of the tasks are sensitive to accuracy and only 10% are sensitive to inference speed. The latter implies that 90% of the tasks are inference speed sensitive and 10% are accuracy sensitive. The simulation results are shown in Fig. 4a. We observe that in scenarios where more tasks are accuracy-sensitive, when the number of tasks reaches 10,000, the total energy consumption of the system is approximately 202,000 joules. This observation suggests that the algorithm, in its pursuit to satisfy stringent accuracy criteria, tends to prefer models with an extensive parameter count, exemplified by the selection of models like LLaMA-33B. Conversely, in situations with more emphasis on the speed of inference, when the number of tasks reaches 10,000, the total energy consumption of the system is approximately 142,000 joules. This means that the algorithm tends to select models characterized by fast inference capabilities, and these models typically have a reduced parameter count, leading to lower energy consumption. A prime example of such a model is LLaMA-7B, which aligns well with the requisites of inference speed-sensitive tasks.

Finally, we focus on comparing the performance of the three algorithms in two different contexts: inference speed-sensitive and accuracy-sensitive scenarios. Fig. 4b illustrates the operation of these algorithms under accuracy-sensitive constraints, while Fig. 4c demonstrates their functionality under inference-speed-sensitive constraints. During the initial stages of iterations, the Epsilon Greedy algorithm performs



(a) UCB in Different Scenarios (b) Accuracy Focus (c) Inference Speed Focus

Fig. 4. Algorithm Performance Analysis

comparable to that of the UCB algorithm. However, there is a clear divergence as the number of iterations increases. The UCB algorithm consistently maintains its optimal performance, demonstrating a more robust and reliable effectiveness over longer iteration sequences, in contrast to the initially competitive but ultimately less consistent performance of the Epsilon Greedy algorithm.

The experiments conducted clearly illustrate the superiority of the UCB algorithm in the context of selecting optimal configuration values for offloading LLM inference tasks. Notably, the UCB algorithm consistently identifies the most suitable configuration value more rapidly than its counterparts, demonstrating an unerring capacity for making accurate decisions. Furthermore, this algorithm exhibits remarkable robustness, as evidenced by its swift convergence across a trio of distinct task requirements.

VI. Conclusion and Future Work

This paper deals with the energy efficiency of the LLM inference task in an edge computing environment. We propose a problem concerning the total energy consumption and task requirement and formulate it as an MAB problem. We address this MAB problem using UCB algorithm. The simulation experiments show that the UCB algorithm can effectively minimize energy consumption and has clear advantages over other algorithms. In future work, we will consider deploying a multimodal large language model in edge environments and observe the configurations that affect the power consumption when reasoning with different modal data.

Acknowledgment

This work is partially supported by JSPS KAKENHI Grant Numbers JP20K11784, JP20H04174, JP22K11989, JP23K11063, JP24K14910, Leading Initiative for Excellent Young Researchers (LEADER), MEXT, Japan, and JST, PRESTO Grant Number JPMJPR21P3, Japan, JST ASPIRE Grant Number JPMJAP2344. This work is also supported by JST SPRING, Grant No. JPMJSP2153, Japan. He Li is the corresponding author.

References

[1] J. Fang, Y. He, F. R. Yu, J. Li, and V. C. Leung, "Large language models (llms) inference offloading and resource allocation in cloud-edge networks: An active inference approach," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pp. 1–5, IEEE, 2023.

[2] D. Luitse and W. Denkena, "The great transformer: Examining the role of large language models in the political economy of ai," *Big Data & Society*, vol. 8, no. 2, p. 20539517211047734, 2021.

[3] M. Zhao, W. Li, L. Bao, J. Luo, Z. He, and D. Liu, "Fairness-aware task scheduling and resource allocation in uav-enabled mobile edge computing networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 2174–2187, 2021.

[4] M. Bolourian and H. Shah-Mansouri, "Energy-efficient task offloading for three-tier wireless powered mobile edge computing," *IEEE Internet of Things Journal*, 2023.

[5] T. Dreibholz and S. Mazumdar, "Towards a lightweight task scheduling framework for cloud and edge platform," *Internet of Things*, vol. 21, p. 100651, 2023.

[6] N. Su, J.-B. Wang, Y. Chen, H. Yu, C. Ding, Y. Pan, and J. Wang, "Joint mu-mimo precoding and computation optimization for energy efficient industrial iot with mobile edge computing," *IEEE Transactions on Green Communications and Networking*, 2023.

[7] A. M. Seid, H. N. Abishu, A. Erbad, and M. Guizani, "Hdfrl-empowered energy efficient resource allocation for aerial mec-enabled smart city cyber physical system in 6g," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 836–841, IEEE, 2023.

[8] M. Ibrar, A. Erbad, M. Abegaz, A. Akbar, M. Houchati, and J. M. Corchado, "Reed: Enhanced resource allocation and energy management in sdn-enabled edge computing-based smart buildings," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 860–865, IEEE, 2023.

[9] S. Zhu, K. Ota, and M. Dong, "Green ai for iiot: Energy efficient intelligent edge computing for industrial internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 1, pp. 79–88, 2021.

[10] A. Galanopoulos, J. A. Ayala-Romero, D. J. Leith, and G. Iosifidis, "Automl for video analytics with edge computing," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10, IEEE, 2021.

[11] X. Yuan, H. Li, K. Ota, and M. Dong, "Building energy efficient semantic segmentation in intelligent edge computing," *IEEE Transactions on Green Communications and Networking*, 2023.

[12] m. S. Abirai and P. Chitra, "Energy-efficient edge based real-time healthcare support system," in *Advances in computers*, vol. 117, pp. 339–368, Elsevier, 2020.

[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[14] W. Zhong, R. Cui, Y. Guo, Y. Liang, S. Lu, Y. Wang, A. Saied, W. Chen, and N. Duan, "Agieval: A human-centric benchmark for evaluating foundation models," *arXiv preprint arXiv:2304.06364*, 2023.