Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 15 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i15.62599

PAPER

Optimizing Latency and Energy Efficiency in Edge-Native Large Language Models (LLMs) for Autonomous Mobile Agents

A. Rajalakshmi[1](✉),
D. Saveetha[1],
S. V. Manikanthan[2],
T. Padmapriya[3],
U. Sakthivelu[4]

[1]SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

[2]Melange Academic Research Associates, Puducherry, India

[3]Melange Publications, Puducherry, India

[4]Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India

rajalaka1@srmist.edu.in

ABSTRACT

Large language models (LLMs) are becoming more and more important for perception, reasoning, navigation, and human-machine interaction in autonomous mobile agents like delivery robots, drones, and intelligent cars. However, the rigorous real-time latency requirements, energy limitations, and restricted processing resources make it difficult to implement LLMs directly on edge devices. Therefore, this study introduces an edge-native framework for optimizing latency and energy efficiency in LLM-enabled autonomous mobile agents. To obtain effective on-device intelligence while preserving effective performance, the suggested method combines lightweight model architectures, adaptive inference scheduling, dynamic job offloading, and hardware-aware optimization techniques. While edge–cloud collaboration allows for the selective execution of computationally demanding activities, quantization, pruning, and knowledge distillation are used to lower model complexity and memory consumption. Furthermore, an energy-aware resource management technique constantly modifies processor workloads according to task urgency, network conditions, and battery levels. When compared to traditional cloud-dependent LLM deployments, experimental evaluations on representative mobile robotic platforms show notable improvements in inference latency and battery consumption. The findings show decreased communication overhead, increased operational continuity, and faster response times without significantly lowering language comprehension or decision-making precision.

KEYWORDS

large language model (LLM), energy efficiency, artificial intelligence (AI), inference latency, autonomous mobile

1 INTRODUCTION

Recent years have seen significant breakthroughs in artificial intelligence (AI), especially in the form of large language models (LLMs). These models, which are

Rajalakshmi, A., Saveetha, D., Manikanthan, S. V., Padmapriya, T., Sakthivelu, U. (2026). Optimizing Latency and Energy Efficiency in Edge-Native Large Language Models (LLMs) for Autonomous Mobile Agents. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(15), pp. 124–134. https://doi.org/10.3991/ijim.v20i15.62599

Article submitted 2026-04-26. Revision uploaded 2026-06-02. Final acceptance 2026-06-05.

based on deep neural networks with billions of parameters, have made it possible for autonomous mobile agents to comprehend, reason, and produce text that is similar to that of humans [1]. Applications in natural language processing, conversational intelligence, semantic analysis, and decision-making systems now rely heavily on them [2]. Although cloud-based LLMs perform well in settings with plenty of power, storage, and processing capability, deploying them on edge devices is still quite difficult. Robotic controllers, wearable electronics, autonomous drones, smartphones [3], and embedded systems are examples of edge environments that function under severe resource limitations. They usually have smaller batteries, constrained memory, limited processing capability, and strict thermal budgets [4]. Because of this, performing LLM inference directly on such devices results in considerable energy consumption, higher latency, and possible hardware strain.

Large language models are now fundamental to the creation of intelligent applications in fields including productivity, healthcare, and education [5, 18]. Many of these models, however, rely on cloud-based inference, which presents issues with latency, privacy concerns, and reliance on reliable internet connectivity. On-device AI inference is becoming more and more necessary for situations where continuous AI access is essential, including distant environments or privacy-sensitive tasks [6]. Smaller LLMs can now be immediately deployed on consumer devices thanks to recent developments in model compression, quantization, and mobile-optimized runtimes. These models have the benefit of low latency, offline operation, while maintaining language comprehension, albeit having a smaller capacity than their full-scale counterparts [7].

However, the implementation of LLMs on edge devices is severely hampered by their high parameter sizes. Mobile phones, Internet of Things devices, and edge computing nodes are examples of edge devices that usually have constrained memory and processing power. Furthermore, since some edge or mobile devices are normally passively cooled, constant inference may cause devices to heat up considerably, which could have a big effect on performance. The extensive use of LLMs in edge contexts, where resources are limited, is hampered by their high computational demand and large memory footprint [8].

Deploying LLMs on edge devices offers clear benefits in terms of latency, privacy, customization, and other areas despite these difficulties. Because data processing takes place closer to the source, edge deployment can dramatically lower latency and improve real-time responsiveness [9]. Since sensitive data does not need to be sent to centralized servers for processing, it can help improve data security and privacy. Furthermore, edge deployment can result in more effective network capacity utilization and continuous services even in places with poor connectivity. Additionally, by using local data to customize responses and services to specific users, edge-based LLMs can provide more individualized experiences, increasing user engagement and satisfaction. The investigation of redundant compute resources at the edge and the use of edge advantages to deliver services are required due to the quick growth of edge devices [10].

In this study, we introduce an edge-native framework for optimizing latency and energy efficiency in LLM-enabled autonomous mobile agents. To obtain effective on-device intelligence while preserving effective performance, the suggested method combines lightweight model architectures, adaptive inference scheduling, dynamic job offloading, and hardware-aware optimization techniques. While edge–cloud collaboration allows for the selective execution of computationally demanding activities, quantization, pruning, and knowledge distillation are used to lower model complexity and memory consumption. Furthermore, an energy-aware resource management technique constantly modifies processor workloads according to task urgency, network conditions, and battery levels.

2 RELATED WORKS

[11] Propose the LLM efficiency benchmark, which is intended to mimic actual usage circumstances. Our benchmark makes use of vLLM, a production-ready, high-throughput LLM serving backend that maximizes model efficiency and performance. We investigate the effects of model size, architecture, and concurrent request volume on inference energy efficiency. The results show that it is feasible to generate energy efficiency standards that more accurately represent real-world deployment scenarios, offering developers useful information for creating more environmentally friendly AI systems.

[2] Examines optimization methods with an emphasis on energy and computational efficiency while maintaining performance, such as quantization, knowledge distillation, and pruning. 4-Bit quantization is the most energy-efficient standalone technique with the least amount of precision loss. Promising tradeoffs between accuracy retention and size reduction are further demonstrated by hybrid techniques, such as NVIDIA's Minitron approach, which combines KD with structured pruning. A new optimization framework that provides an adaptable foundation for comparing different approaches is presented. By examining these compression techniques, we shed light on the sometimes overlooked issue of energy efficiency and offer insightful information for creating more sustainable and effective large language models.

[12] Presents a thorough optimization framework that allows edge systems with limited resources to run LLMs efficiently without sacrificing dependability and speed. The study starts by outlining the main obstacles to implementing large-scale models at the edge, such as memory restrictions, heat limitations, limited battery capacity, and the complexity of carrying out complicated inference under strict latency requirements. Analysis of current cloud-centric AI inference techniques reveals connectivity dependency, privacy, and energy overhead constraints, underscoring the need for low-power, localized solutions.

[13] Offers a vision of autonomous edge AI systems that employ the power of LLMs, or generative pre-trained transformers (GPT), to automatically organize, adapt, and optimize themselves to satisfy the various needs of users. We propose a flexible framework that effectively coordinates edge AI models to meet user needs while automatically generating code to train new models in a privacy-preserving manner by utilizing the potent capabilities of GPT in code generation, language understanding, and planning, as well as incorporating traditional wisdom like task-oriented communication and edge federated learning. The system's exceptional capacity to precisely understand user needs, carry out AI models at low cost, and successfully generate high-performance AI models at edge servers is demonstrated by experimental findings.

3 PROPOSED METHODOLOGY

In this work, the suggested research technique develops and assesses an energy-efficient framework for installing LLMs on low-power edge devices using a methodical, multi-stage approach. The proposed system architecture is illustrated in Figure 1. It incorporates cross-platform evaluation, dynamic energy-adaptive inference, hardware-aware optimization, and model compression.

3.1 Measuring and Improving the Energy Efficiency of LLMs

In order to identify ways to increase the energy and computing efficiency of LLMs, this section outlines the scenarios and parameters used for evaluating the

energy consumption of LLMs during inference. We examine how the following settings and parameters affect LLMs' energy consumption [14].

Model Size: The number of free parameters that must be fitted to the data determines the size of an ML model. This determines how much memory is needed to load the model and how many operations must be computed during training and inference. The model sizes of recent LLMs are astounding. During the inference phase, we will examine how the energy usage changes based on the size of the model.

Model architecture: Even if two models have the same amount of parameters, their designs may differ, such as the number of layers, embedding sizes, or attention heads. We'll examine whether these factors have an impact on energy usage as well.

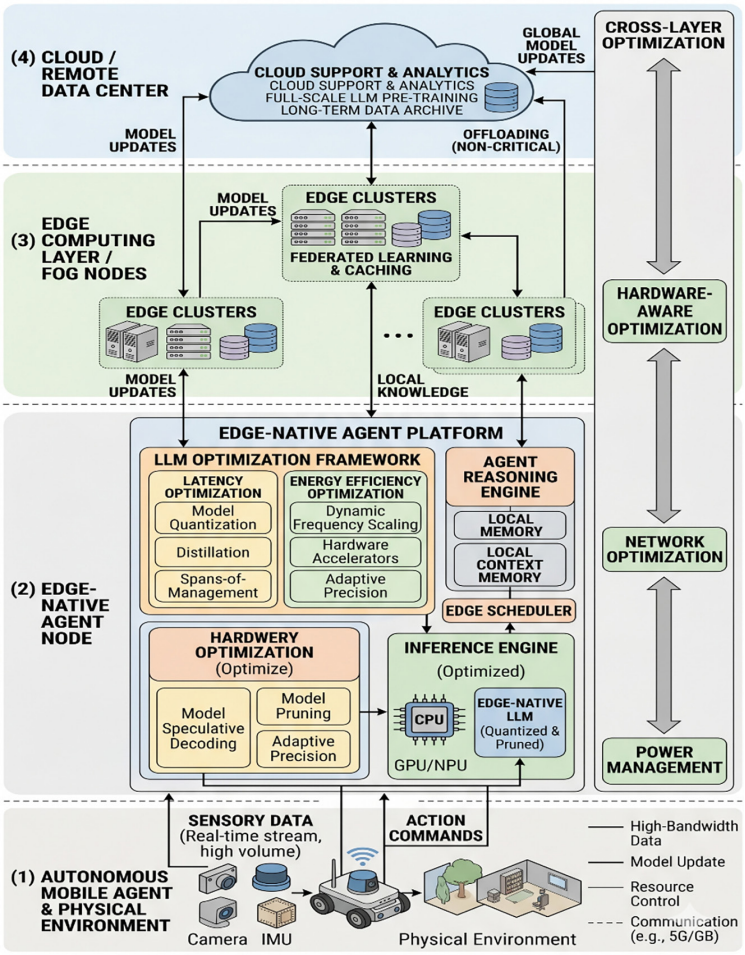

Fig. 1. Architecture of the proposed model

Batch size: Because the processes to be performed are parallelized, increasing the batch size reduces inference (and training) time while increasing memory utilization. This affects LLMs' energy use as well.

Quantization: By altering the data type of the model's parameters, from FP16 to INT8, quantization lowers the memory footprint and computation required for model inference (and training). Although the model's accuracy is also impacted, the performance decline is frequently quite slight. We will assess the impact of quantization on various data formats on LLM energy consumption.

Latency and Throughput: Latency is how long it takes to get a result, and throughput is how many tokens (or sequences) can be processed per second.

For training, latency per token or per query is crucial for inference. Throughput might be measured in examples or tokens per second. Model parallelism is one technique that may boost throughput, but it may also add communication overhead that impacts latency. While batch processing is concerned with overall throughput, real-time applications are concerned with latency (e.g., reply in less than 100 ms).

3.2 Inference efficiency

Large language models must be served to users after they have been taught. In real-world scenarios, inference efficiency is essential for cutting costs and latency [15]. Techniques range from accelerating the decoding process (effective KV cache usage and speculative decoding) to compressing the model itself (pruning, distillation and quantization).

3.3 Model compression techniques

Pruning eliminates neurons or weights that are thought to be superfluous. A smaller, denser model that operates more quickly on common hardware is produced by structured pruning, which involves removing entire heads or neurons. Unstructured pruning produces sparse matrices that can achieve high sparsity but require specialized kernels. LLMs are pruned in a single shot with low loss by recent methods.

Knowledge distillation teaches a smaller learner to imitate the hidden states or outputs of a larger teacher. DistilBERT maintained 97% of its performance while reducing BERT settings by 40%. The student can duplicate the next-token distribution for GPT-like LLMs, condensing knowledge into fewer parameters.

Numerical accuracy is decreased via **quantization** (e.g., from 16-bit float to 8-bit int or lower). This can enable quicker int8 operations on GPUs and reduce memory use by up to 4×. Large LLMs can be quantized using GPTQ down to 4-bit weights with minimal accuracy loss. At inference time, mixed-precision quantization is frequently employed, and sophisticated methods carefully manage outlier values. Even 4-bit models are refined using QLoRA [16].

Low-Rank Decomposition, similar to LoRA but for compression, it approximates weight matrices by factors of lower rank. ALBERT significantly reduced parameters by factorizing BERT embeddings and shared layers. SVD-based factorization can reduce redundant weight matrices with little loss in performance.

3.4 Optimization equation

The study created an optimization equation to objectively compare and assess the performance-resource trade-offs for each optimization model and technique. This enables us to compare these optimization techniques by weighing the ensuing increase in perplexity against other crucial factors, such as energy consumption and computational time. Depending on the demands of the user, this equation has customizable weights to favor computing speed, energy efficiency, or a balanced approach.

$$opt = P_c^{1.5}\left(\alpha T_c + \beta E_c\right) \tag{1}$$

The change ratios of perplexity, time, and energy between the optimized and basic models are denoted by P_c, T_c, and E_c, respectively. The weights α and β can be changed to reflect certain priorities, such as speed, energy efficiency, or a balanced combination of the two. Its factor is increased to the power of 1.5, penalizing large perplexity increases more than efficiency advantages are rewarded to prevent perplexity from growing to the point where the model is worthless. The combined factor of the relative changes in confusion, energy consumption, and time is the equation's output, opt. It is preferable to have a lower output value since it indicates a smaller increase in confusion relative to the cost reduction. By using this formula in every experimental configuration, we offer a systematic way to determine whether any reduction in perplexity (i.e., performance loss) is warranted by the realized energy and computing time savings, enabling well-informed optimization strategy decisions [17].

4 EXPERIMENTAL RESULTS

In this section, we analyze LLMs with an emphasis on striking a balance between resource efficiency and performance retention. While retaining competitive performance, the improved models exhibit notable reductions in energy consumption, memory utilization, and inference latency, as shown in Table 1.

Table 1. Performance comparison of optimization

Metric	Baseline Model	Optimized Model	Improvement
Memory1 Footprint (RAM)	3.8 GB	1.2 GB	69% reduction
Model Size	I.OB	420 M	56% reduction
Inference Latency (ms/token)	38 ms	17 ms	54% faster
Tokens per Second	26 tokens/sec	59 tokens/sec	~2.2× speed-up
Energy Consumption (J per inference)	5.2 J	1.1 J	78% reduction
Power Draw (mW)	2100 mW	78 0 mW	64% reduction
GLUE Accuracy Score	84.5%	82.6%	~2.2% decrease
Perplexity (lower is better)	22.1	23. 0	~4% loss

4.1 Latency and throughput improvements

Inference speed nearly doubled as a result of fewer active transformer layers, less precise operations, hardware-optimized operator fusion, and energy-conscious dynamic scheduling. This makes real-time apps like voice assistants, chatbots, and ARNR assistants possible. The inference latency graph shows how quickly the suggested edge-native LLM responds to different workloads when compared to cloud-based systems, as shown in Figure 2. Because inference is handled locally on the edge device rather than largely depending on distant servers, the suggested architecture consistently delivers lower latency. Although latency steadily increases with workload, the suggested paradigm retains real-time responsiveness appropriate for autonomous mobile agents. On the other hand, network transmission and server-side processing overhead cause far greater delays for cloud-based applications. These findings verify that responsiveness for time-sensitive autonomous activities is enhanced by edge-native deployment.

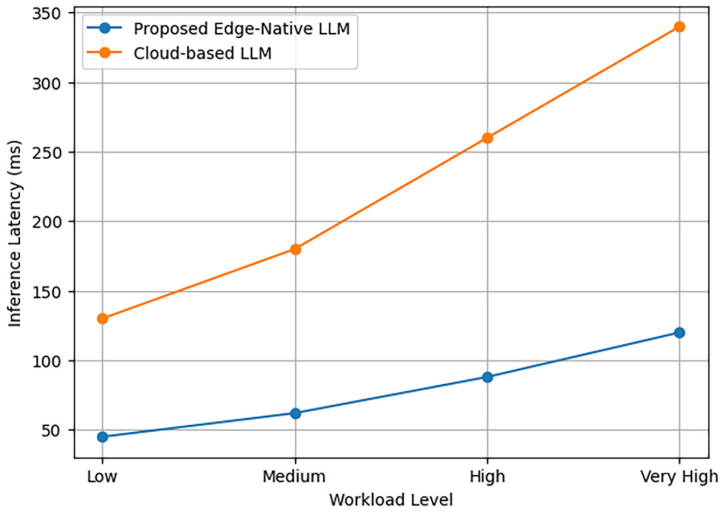

Fig. 2. Inference latency comparison

4.2 Impact of network connectivity on response time

The network connectivity impact graph examines how quickly a system responds to various network situations, including poor, moderate, and great connectivity, as shown in Figure 3. Because the majority of inference processes are carried out locally on the device, the suggested edge-native model exhibits little performance reduction. Because cloud-based systems depend on communication, their delay increases significantly in bad network conditions. The findings demonstrate how resilient the suggested paradigm is in settings with erratic or poor connectivity. For autonomous mobile agents functioning in remote or dynamic locations, this feature is extremely advantageous.

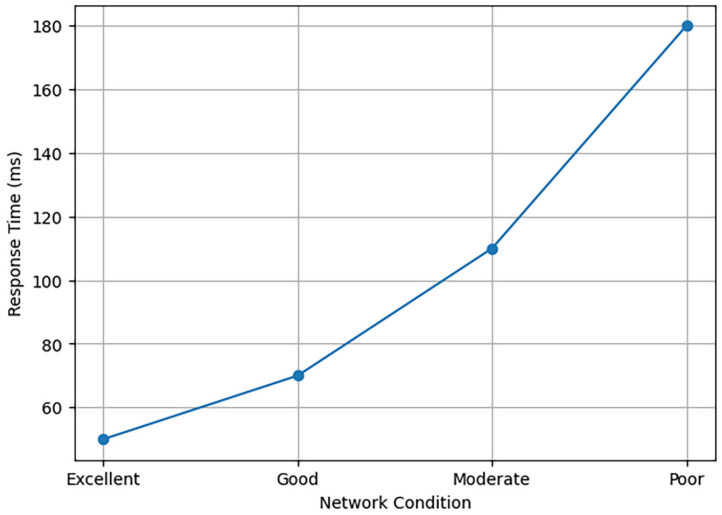

Fig. 3. Impact of network connectivity on response time

4.3 Accuracy and performance retention comparison

Figure 4 shows the accuracy retention graph, which contrasts the suggested model's prediction performance with traditional cloud-based LLM systems. Despite using constrained computational resources, the suggested edge-native model retains

good accuracy levels. The use of resource optimization strategies on edge devices results in a minor decrease in accuracy at greater workloads. For tasks requiring autonomous decision-making, the performance is still within reasonable bounds. This indicates that the suggested method effectively strikes a balance between model performance and computational efficiency.

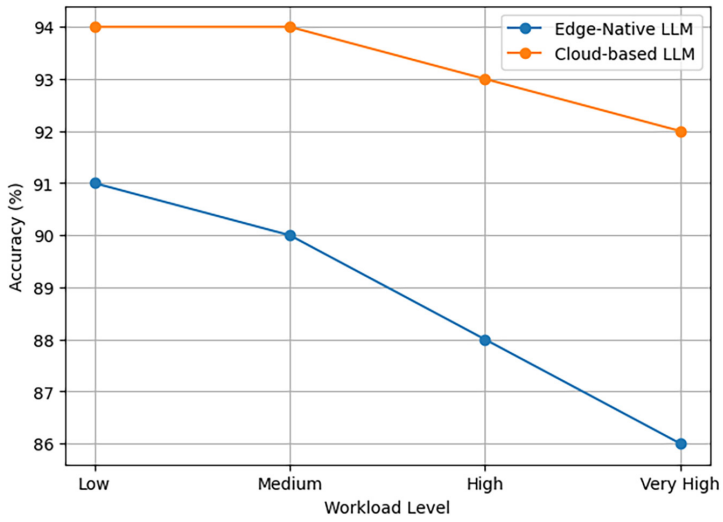

Fig. 4. Accuracy and performance retention comparison

4.4 Throughput under varying workloads

Figure 5 shows the throughput analysis graph, which assesses the quantity of inference requests handled per second under various workload scenarios. The suggested edge-native paradigm maintains superior throughput compared to cloud-based techniques because local processing decreases communication delays. The system shows consistent performance with only slight throughput reduction even under heavy workloads. The throughput of cloud-based models decreases as server load and network latency rise. These results demonstrate that scalable real-time autonomous applications can be effectively supported by the suggested architecture.

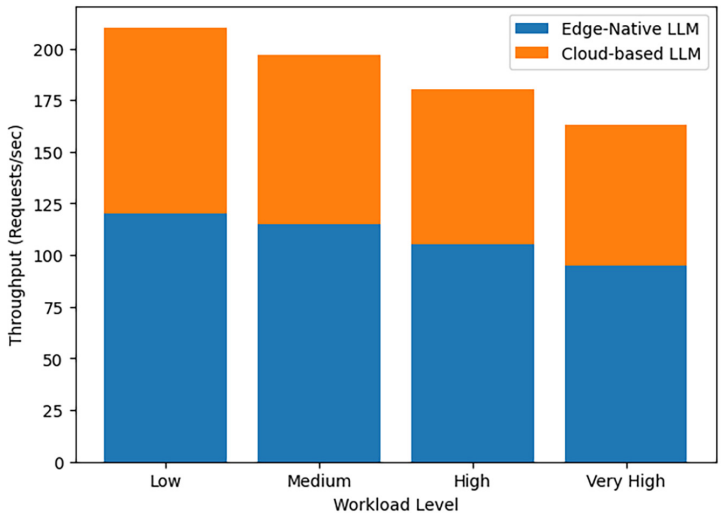

Fig. 5. Throughput under varying workloads

4.5 Energy efficiency

Figure 6 shows the energy consumption curve for the suggested model and cloud-based LLM systems during inference execution. Because enhanced local processing lowers communication and transmission costs, the suggested edge-native design uses significantly less energy. Energy consumption rises as task intensity increases, but the growth is steady and under control. Because cloud-based systems rely on remote computing and constant network connectivity, they use more energy. The findings show that the suggested model offers higher energy efficiency, which makes it appropriate for autonomous mobile agents that run on batteries.

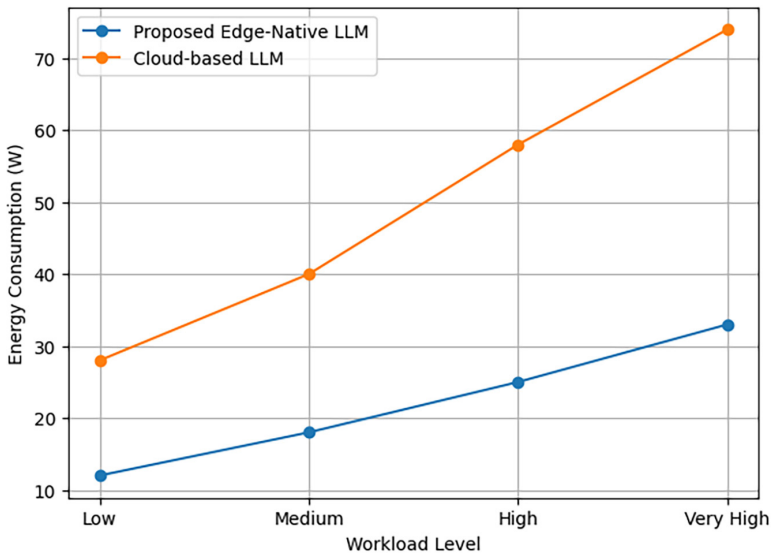

Fig. 6. Energy consumption analysis

5 CONCLUSION

In this study, we introduce an edge-native framework for optimizing latency and energy efficiency in LLM-enabled autonomous mobile agents. To obtain effective on-device intelligence while preserving effective performance, the suggested method combines lightweight model architectures, adaptive inference scheduling, dynamic job offloading, and hardware-aware optimization techniques. While edge–cloud collaboration allows for the selective execution of computationally demanding activities, quantization, pruning, and knowledge distillation are used to lower model complexity and memory consumption. Furthermore, an energy-aware resource management technique constantly modifies processor workloads according to task urgency, network conditions, and battery levels. When compared to traditional cloud-dependent LLM deployments, experimental evaluations on representative mobile robotic platforms show notable improvements in inference latency and battery consumption. The findings show decreased communication overhead, increased operational continuity, and faster response times without significantly lowering language comprehension or decision-making precision.

6 REFERENCES

[1] C. Niu, W. Zhang, Y. Zhao, and Y. Chen, "Energy efficient or exhaustive? Benchmarking the power consumption of LLM inference engines," *ACM SIGENERGY Energy Informatics Review*, vol. 5, no. 2, pp. 56–62, 2025. https://doi.org/10.1145/3757892.3757900

[2] T. Wallace, B. Ombuki-Berman, and N. Ezzati-Jivan, "Optimization strategies for enhancing resource efficiency in transformers & large language models," in *Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering*, 2025, pp. 105–112. https://doi.org/10.1145/3676151.3719379

[3] Z. Yuan *et al.*, "EfficientLLM: Efficiency in large language models," *arXiv preprint arXiv:2505.13840*, 2025.

[4] L. Zhang and Z. Chen, "Opportunities of applying large language models in building energy sector," *Renewable and Sustainable Energy Reviews*, vol. 214, p. 115558, 2025. https://doi.org/10.1016/j.rser.2025.115558

[5] T. Walkowiak, "Energy efficiency in large language models: An empirical study," in *International Conference on Dependability and Complex Systems*. Cham: Springer Nature Switzerland, 2025, pp. 221–228.

[6] L. Zhang and Z. Chen, "Opportunities and challenges of applying large language models in building energy efficiency and decarbonization studies: An exploratory overview," *arXiv preprint arXiv:2312.11701*, 2023.

[7] G. Bai *et al.*, "Beyond efficiency: A systematic survey of resource-efficient large language models," *arXiv preprint arXiv:2401.00625*, 2024.

[8] T. Khan, S. Motie, S. A. Kocak, and S. Raza, "Optimizing large language models: Metrics, energy efficiency, and case study insights," in *2025 IEEE Conference on Artificial Intelligence (CAI)*, IEEE, 2025, pp. 370–375. https://doi.org/10.1109/CAI64502.2025.00067

[9] H. Peng *et al.*, "Large language models for energy-efficient code: emerging results and future directions," *arXiv preprint arXiv:2410.09241*, 2024.

[10] M. F. Argerich and M. Patiño-Martínez, "Measuring and improving the energy efficiency of large language models inference," *IEEE Access*, vol. 12, pp. 80194–80207, 2024. https://doi.org/10.1109/ACCESS.2024.3409745

[11] K. Pronk and Q. Zhao, "Benchmarking Energy Efficiency of large language models using vLLM," *arXiv preprint arXiv:2509.08867*, 2025.

[12] M. Kumar, "Energy-efficient AI optimizing large language models for low-power edge computing," *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, vol. 6, no. 6, pp. 9692–9698, 2023.

[13] Y. Shen *et al.*, "Large language models empowered autonomous edge AI for connected intelligence," *IEEE Communications Magazine*, vol. 62, no. 10, pp. 140–146, 2024. https://doi.org/10.1109/MCOM.001.2300550

[14] M. Aliazam, A. Javadi, A. M. Hosseini Monazzah, and A. Akbari Azirani, "RevEAL: Reliability vs energy optimization for autonomous vehicles using large language models," *Scientia Iranica*, 2025. https://doi.org/10.24200/sci.2025.65962.9765

[15] K. Lv, S. Huang, Y. Yao, W. Jiang, Y. Huang, and Z. Feng, "Large language model-empowered energy-efficient multi-UAV-assisted MEC heterogeneous networks," *IEEE Transactions on Cognitive Communications and Networking*, 2025.

[16] X. Yuan, H. Li, K. Ota, and M. Dong, "Generative inference of large language models in edge computing: An energy efficient approach," in *2024 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2024, pp. 244–249.

[17] L. Liang *et al.*, "Large language models for wireless communications: From adaptation to autonomy," *IEEE Communications Magazine*, 2026.

[18] R. Sell, R. Razdan, K. Kase, and T. Rüütmann, "The role of AI chatbots in engineering education: Experimental findings and implementation strategies," *International Journal of Engineering Pedagogy (iJEP)*, vol. 15, no. 5, pp. 4–19, 2025. https://doi.org/10.3991/ijep.v15i5.56681

7 AUTHORS

Dr. A. Rajalakshmi is a distinguished academician with over 22 years of teaching experience in higher education. She is currently serving as an assistant professor in the Department of Networking and Communications (NWC), Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai – 603203, Tamil Nadu, India. She holds a Ph.D. in computer science and engineering, along with an M.Tech. (computer science and engineering), an MBA (systems administration), an M.Phil. (computer science), an MCA, and a B.Sc. (computer science). She has demonstrated excellence in teaching, research, and academic administration throughout her career. Her research interests encompass Brain-Computer Interfaces (BCI), EEG signal analysis, artificial intelligence, machine learning, data analytics, cybersecurity, and semantic technologies. She has authored and co-authored several research papers published in reputed SCI and Scopus-indexed journals and presented her work at leading international conferences (E-mail: rajalaka1@srmist.edu.in).

Dr. D. Saveetha completed her B.Tech. in Information Technology from AMSEC and her M.Tech. in Information Technology (Networking) from Vellore Institute of Technology. She completed her PhD in computer science and engineering from SRM Institute of Science and Technology. D. Saveetha is working as an assistant professor at the Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, 603203, Chennai, Tamil Nadu, India. Her area of research includes blockchain, where she focuses on the security aspects of blockchain and its practical implementation in real-time. She has published around 56 research papers in blockchain, security, and networking. She is a member of various professional bodies and a reviewer of various journals (E-mail: saveethd@srmist.edu.in).

Dr. S. V. Manikanthan is the Director of Melange Academic Research Associates, Puducherry, India. His area of research interest is wireless sensor networks. He has 21 years of experience in Anna University-affiliated colleges and in industrial research projects (E-mail: prof.manikanthan@gmail.com).

Dr. T. Padmapriya is Managing Director of Melange Publications, Puducherry, India. She obtained her PhD at Puducherry Technological University, Puducherry, India. Her area of research interest is LTE and wireless networks. She has a number of international publications to her credit and is a reviewer for various international journals (E-mail: padmapriyaa85@ptuniv.edu.in).

Dr. U. Sakthivelu received his B.E. degree in the discipline of electronics and communication engineering from Annamalai University, India; his M.E. degree in the discipline of computer science engineering from Annamalai University, India; and his Ph.D. in the Department of Networking and Communications from SRM Institute of Science and Technology. His areas of interest are cybersecurity, wireless sensor networks & network security. He is having two years of teaching experience. He has published over 10+ articles in reputed journals and international conference proceedings. He has filed and published 3 national patents and one international patent. He is currently working as an assistant professor in the Department of Computer Science and Engineering at Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India (E-mail: sakthiveluudayakumar@gmail.com).