Deploying Large Language Model on Cloud-Edge Architectures: a case Study for Conversational Historical Characters

Beniamino Di Martino, Mariangela Graziano, Luigi Colucci Cante

Abstract This work analyzes the deployment of conversational agents based on large language models (LLMs) in cloud-edge architectures, placing emphasis on scalability, efficiency and real-time performance.
Through a case study, we present a web application that allows users to interact with an augmented reality avatar that impersonates a historical character. The agent, powered by an LLM delivers immersive and contextually coherent dialogues.
We discuss the solutions adopted to manage latency and distribute the computational load between the cloud, which takes care of language processing, and the edge nodes, ensuring a smooth user experience.
The results obtained demonstrate how accurate design can optimize the use of LLMs in distributed environments, offering advanced and high-performance interactions even in applications with high reactivity and customization requirements.

1 Introduction

The application of Large Language Models (LLMs) has impacted numerous fields, particularly conversational agents, which can now enable fluid, natural, and contextually consistent interactions.
However, its application in distributed environments, such as cloud-edge architectures, presents a number of opportunities and concerns that require careful consid-

Mariangela Graziano e-mail: mariangela.graziano@unicampania.it
Department of Engineering, University of Campania "L. Vanvitelli", Aversa, Italy

Luigi Colucci Cante e-mail: luigi.coluccicante@unicampania.it
Department of Engineering, University of Campania "L. Vanvitelli", Aversa, Italy

Beniamino Di Martino e-mail: beniamino.dimartino@unicampania.it
Department of Engineering, University of Campania "L. Vanvitelli", Aversa, Italy
Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan
Department of Computer Science, University of Vienna, Vienna, Austria

eration. When these models are employed in cloud-edge architectures, significant benefits emerge, but also some challenges to be addressed [18].
One particularly interesting aspect concerns the execution of LLMs on-device. In this scenario, processing occurs directly on the user's device, reducing latency and improving the real-time experience. This approach is particularly advantageous in applications such as conversational agents powering an augmented reality avatar, where immediate interaction is crucial. Furthermore, local data processing increases privacy, preventing sensitive information from being transferred to remote servers, thus reducing the risk of unwanted exposure [10, 2].
It is important to highlight that when processing occurs exclusively in the cloud, some security and privacy issues arise [4, 6]. Data sent to the cloud is vulnerable to possible attacks or unauthorized access [1]. Furthermore, even though the cloud has huge computational resources, it may not be able to meet the real-time performance needs of highly dynamic applications, such as those requiring constant interaction. The cloud-edge solution seems to be the most promising, since it combines the computational power of the cloud with the advantages of local processing [28].
In this work, we explore how to balance the load between the cloud and edge devices to achieve an optimal user experience, while ensuring the scalability and security needed in highly personalized and reactive scenarios.

2 Related Work

Large language models are very powerful for natural language understanding and generation, but they require a lot of computational resources and memory, which often exceed the capabilities of edge devices.
In this section, we report some works that discuss the deployment of large language models in cloud and edge platforms.
The works [31, 20] explore techniques to make LLMs more efficient, such as model compression before their deployment and optimizations during execution. Furthermore, [31] discusses practical applications of these models on edge devices in various domains, such as personal, enterprise and industrial use, thus providing a comprehensive overview of state-of-the-art methodologies for the deployment of LLMs on resource-constrained devices, bridging the gap between their huge potential and the limitations of edge computing. To address the growing size and computational demands of LLMs, which are becoming increasingly difficult to manage on limited hardware or via remote cloud, in work [19] a new paradigm called Mobile Edge Intelligence (MEI) is proposed, which, by positioning itself between devices and the cloud, offers greater computing power than end devices, but with advantages in terms of privacy and low latency compared to the cloud. The work opens the reflection on the fact that the cloud is no longer sufficient to meet the increasingly stringent requirements of latency, bandwidth and privacy required by modern applications using LLMs and at the same time considers MEI as an ideal compromise, but requires innovative solutions to overcome the hardware and resource limitations

of the mobile edge. The authors also highlight the importance of integrating artificial intelligence with 6G to support future scenarios in which LLMs will be even more central to our lives.
The articles [15, 25] are part of the ongoing research on optimizing LLM deployment in distributed environments, addressing the challenges related to latency and computational demands of cloud-edge systems. The first proposes an innovative framework called CE-CoLLM, which combines autonomous edge inference and cloud-edge collaboration to ensure efficiency and high accuracy. The solution relies on techniques such as early-exit mechanism, cloud context manager and quantization to reduce latency and costs. CE-CoLLM represents a step forward towards more sustainable LLMs suitable for latency-, privacy- and cost-sensitive applications. The second proposes a comprehensive review of the most advanced techniques of optimization of LLMs for execution on edge devices, including model compression (quantization, pruning, distillation), efficient architectural designs and hardware-software co-optimization strategies. This approach points out that the transition from cloud-centric to edge-based models allows greater customization and energy cost reduction, while addressing the challenges related to resource limitations of edge devices. The research highlights the need for multidisciplinary efforts to achieve a balance between performance, sustainability and security, positioning on-device models as a key element for a more accessible and responsible AI.
Regarding the line of reflection on the topic of running LLMs models on mobile devices, related to the exploration of the possibilities of integrating advanced computational capabilities and increasing privacy needs, work [16] presents an automation infrastructure called MELT, which is a platform for headless execution and benchmarking of LLMs on devices such as Android, iOS and Nvidia Jetson. This infrastructure allows to analyze the performance, memory and energy consumption of different models, evaluating their efficiency in real contexts. The work highlights that the execution of LLMs on mobile devices is strongly limited by memory consumption and energy impact, making their continuous execution impractical due to negative effects on the user experience and that Quantization proves effective in reducing memory requirements but entails a non-negligible loss of precision. Furthermore, a significant heterogeneity in performance emerges between the various devices and frameworks.
According to [12], the evolution of advanced driver assistance systems (ADAS), trying to overcome the limitations of traditional approaches, which are unable to fully understand the surrounding environment or to effectively integrate data from multiple sensors. It proposes a collaborative system based on multimodal language models (MLLMs), which exploits a combination of cloud and edge computing. In particular, the lighter CogVLM2 model is used at the edge for rapid response, while ChatGPT-4o in the cloud handles more complex tasks. The main results highlight how model fine-tuning and the use of deep learning techniques significantly improve decision-making capabilities in traffic scenarios, while the task offloading algorithm optimizes latency, energy consumption and quality of service. This approach demonstrates a significant advance over traditional methods, paving the way for even more intelligent and adaptive driver assistance.

3 Deployment Models: Cloud, Edge, and Distributed Cloud-Edge

Regarding the implementation of artificial intelligence models, such as LLM, deployment strategies are traditionally divided into two main approaches: cloud and edge, with the evolution towards the distributed cloud-edge model representing a synthesis of their advantages [22].
In the **cloud model**, the entire processing process takes place on centralized servers. Data is transmitted to the cloud and processed utilising strong and scalable technology before being returned to the user. This method can handle even the most complex models while utilising almost unlimited computer resources. However, it is heavily reliant on a consistent and fast internet connection, making it unsuitable in environments with weak connectivity. Furthermore, regularly moving data to the cloud can create issues about latency and privacy, particularly for applications that handle sensitive data [26].
The **edge model**, on the other hand, relies on local processing, directly on edge devices, such as smartphones, IoT sensors, or gateways [5]. This approach reduces latency, since data does not need to be transmitted to the cloud, and improves privacy by keeping information directly on the device [7]. However, it is limited by the computational power of local hardware, which is often not enough to run large models or intensive processing in real time. As a result, this approach is best suited for lightweight applications or contexts where privacy is a priority [11].
The **distributed cloud-edge model** [8] combines the strengths of the two approaches. Here, the cloud handles the more expensive tasks, such as large-scale data processing or training complex models, while edge devices take care of lighter or more localized operations, such as data pre-processing or running fast inferences. This balance reduces latency, as some processing happens close to the user, and improves privacy, by keeping sensitive data on the local device when possible. Additionally, the edge can provide business continuity even in the event of a temporary disconnect from the cloud, improving overall resilience.
Compared to the cloud-only approach, the cloud-edge model offers lower latency and greater privacy protection, while maintaining the ability to handle complex workloads with the support of the cloud. On the other hand, compared to the edge-only approach, the distributed model overcomes local computational limitations, leveraging the power of the cloud without completely sacrificing proximity to the device. The cloud-edge model represents a more balanced and versatile solution, suitable for modern scenarios where latency, privacy, and processing capacity must coexist harmoniously [29].

4 Proposed Architecture

Figure 1 shows the architecture proposed in this work.

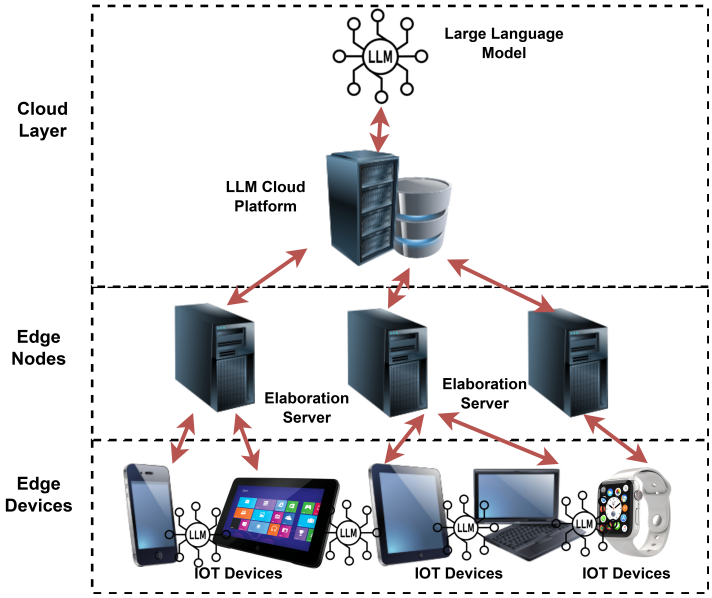


Fig. 1: The Proposed Architecture

As can be seen from Figure 1, the architecture was structured in the three layers typical of the cloud edge paradigm, detailed below:

- **Edge Devices**: users can use their personal devices, such as mobile phones, tablets, computers and smartwatches to make user queries to the conversational agent. By means of LLMs optimised for devices with few computing resources, such as *Mistral* [14], *TinyGPT* [27], *DistilBERT* [21] or *GPT-J* [17], a response to the user query is already produced, but is first forwarded to the Elaboration Servers.
- **Edge Nodes**: the Elaboration Servers receive the user query, response pair from the edge devices, and perform a semantic similarity search from a dataset of questions/answers prepared by domain experts, using **Word Embedding** techniques, to verify whether the user query is present in the dataset [9]. If so, the domain expert's reference answer is retrieved, and it is compared with the answer generated by the LLM of the Edge Devices level to check how correct it is. Metrics such as **BERT Score** [30] can be used for this task. If the score obtained is too low, the Elaboration Server requests the LLM Cloud Platform to direct the user query to a much more powerful cloud-deployed LLM. In the case that the semantic similarity search fails, or if the score obtained is sufficiently high, the Elaboration Server instructs the Edge Device that invoked it to return the answer provided by the LLM at the Edge Device level to the user via a positive ACK message.
- **Cloud Layer**: if the score calculated by an Elaboration Server is lower than a set threshold, it means that the LLM response of the Edge Devices layer was

not very accurate. Then, the user query is forwarded to an LLM Cloud Platform, which uses a much more powerful LLM such as, for example, *Gemini* [23] or *GPT-4* [3] to obtain a much more accurate answer, which is then returned to the final user.

The advantage of this architecture is that most of the requests are handled by the LLMs deployed on the devices, and only those that require better accuracy in their responses are forwarded to the Cloud layer. This balances the tradeoff between speed and accuracy of response. The main problem with this approach is the cost, as it is not trivial to build datasets with the answers for the most important questions in a given domain. Therefore, this approach becomes particularly effective when the context is limited to dialogue with conversational agents who must possess limited knowledge, such as that of a historical character.

5 Implementation

Figure 2 shows the implementation of the approach defined in Section 4, represented using a UML Component Diagram.

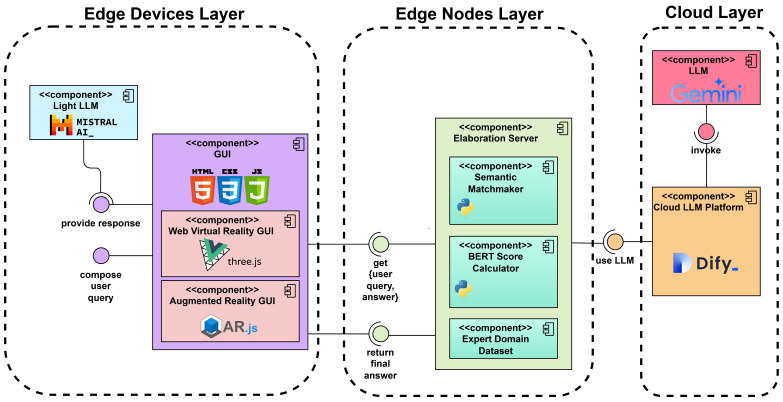


Fig. 2: Implementation of the Proposed Architecture

As can be seen from Figure 2, components have been defined for each layer. In the "*Edge Devices Layer*" there is a "*GUI*" component made with "**HTML**", "**CSS**" and "**Javascript**" Web technologies. It contains special components to enable enjoyment of the application in Virtual Reality Web and Augmented Reality mode throught a 3d avatar, made with **Three.JS** and **AR.js** technologies. This layer allows a request to be made to a light LLM; the **Mistral** model [24] was chosen. In the "*Edge Node Layer*" there is an "*Elaboration Server*" component, which consists of two main sub-components: a "*Semantic Matchmaker*" and a "*BERT Score*

Calcuator", both implemented in **Python**. The Word Embedding model used is **RoBERTa**[1]. Finally, the "*Cloud Layer*" contains a "*Cloud LLM Platform*" component, implemented with **Dify**[2]. Through the Dify platform access is made to the **Gemini** model [13], deployed in the Cloud.

6 A Case Study and Results

In order to test the approach described in Section 4, a web application that allows users to talk to conversational agents using LLM to respond was implemented.
The conversational agent realised in this case study simulates the knowledge and behaviour of the historical character "*Plinio il Vecchio (Pliny the Elder)*"[3]. A group of four historians created a dataset with approximately 50 reference answers to be used to verify the correctness of the application's answers.

Figure 3 shows the graphic interface of the application to dialogue with the historical character, shown with a 3D avatar.

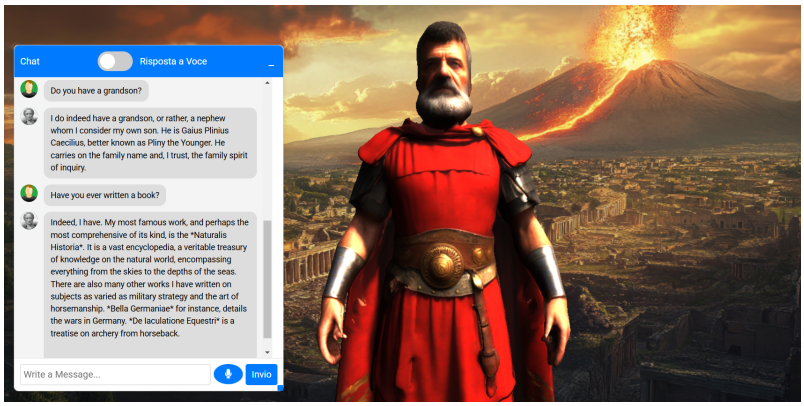


Fig. 3: Graphical interface for conversational agent in Virtual Web Mode

As can be seen from Figure 3, the conversational agent, either through a Virtual Reality Web chat or through a text to speech and speech to text system, is able to respond with quality answers to the end user. Table 1 shows the percentage of

[1] https://huggingface.co/docs/transformers/model_doc/roberta

[2] https://dify.ai/

[3] Gaius Plinius Secundus (AD 23/24–79), known as Pliny the Elder was a Roman author, naturalist, natural philosopher, and naval and army commander of the early Roman Empire, and a friend of the emperor Vespasian. He wrote the encyclopedic Naturalis Historia, a comprehensive thirty-seven-volume work covering a vast array of topics on human knowledge and the natural world, which became an editorial model for encyclopedias. He spent most of his spare time studying, writing, and investigating natural and geographic phenomena in the field.

responses provided with the light LLM Mistral and those provided with the LLM Gemini deployed in the Cloud, calculated from a sample of 1000 responses.

Table 1: Average results obtained from a sample of 1000 uses of the system

LLM	**Percentage of Responses**	**Average BERT Score**	**Average Time of Response**
Mistral	37	0.35	621ms
Gemini	63	0.71	1247ms

As can be seen from the Tables 1, about 37% of the responses were provided via Light LLM Mistral, reducing the latency of the response. The remaining 63% of the responses were provided by the Gemini model deployed in Cloud, as the BERT Score calculated by comparing the response provided by the Mistral LLM and the domain expert's reference response did not exceed the 65% threshold. The system was tested by several users using a web app; the Cloud Edge architecture was of great help in reducing response latency while ensuring the quality of the generated responses, as: i) Mistral's inaccurate answers to important questions were corrected by the Gemini LLM, so that the power of Gemini ensured higher accuracy at the expense of response time; ii) questions that were well answered by Mistral or questions for which domain experts did not provide a reference answer were answered by Mistral, reducing the traffic of requests to the Gemini LLM.

7 Conclusion and Future Works

This work has demonstrated how the adoption of distributed cloud-edge architectures can represent an effective solution for the deployment of conversational agents based on LLMs. Through the case study of a web application integrating an augmented reality avatar, capable of impersonating a historical character, we have highlighted the benefits of a balanced design: the cloud manages linguistic processing, ensuring the necessary computational power, while edge nodes improve responsiveness and preserve user privacy. This approach has allowed to obtain immersive, contextually coherent and performant interactions, even in scenarios with high demands for personalization and responsiveness.
However, the work is not without challenges. The optimal distribution of loads between cloud and edge requires significant design complexity, especially to guarantee minimal latency without overloading the local hardware. Furthermore, an additional level of care is needed to ensure that the dialogues provided by the avatar are historically accurate, avoiding anachronisms or biases that could compromise the reliability of the system in educational and cultural contexts.
Looking ahead, the system can be expanded to include a wider range of historical characters, thus enriching the user experience and offering new opportunities

for learning and entertainment. Improvements in resource efficiency and balancing latency will be key to further optimizing the system performance. Furthermore, the exploration of other application domains, such as literary or mythological characters, will open new frontiers for the creative and educational use of cloud-edge architectures.

Acknowledgements The work described in this article has been supported by the research projects: RASTA - Realtà Aumentata e Story-Telling Automatizzato per la valorizzazione di Beni Culturali ed Itinerari; Italian MUR PON Proj. ARS01 00540 and by the research project SERICS (PE00000014), under the MUR NRRP funded by the EU - NextGenerationEU.

This preprint has not any post-submission improvements or corrections. The Version of Record of this contribution is published in Advanced Information Networking and Applications, and is available online at https://doi.org/10.1007/978-3-031-87778-0_19

References

1. Joseph Abrera et al. Data privacy and security in cloud computing: A comprehensive review. *Journal of Computer Science and Information Technology*, 1(1):01–09, 2024.
2. Mahdi Alkaeed, Adnan Qayyum, and Junaid Qadir. Privacy preservation in artificial intelligence and extended reality (ai-xr) metaverses: A survey. *Journal of Network and Computer Applications*, page 103989, 2024.
3. Jawid Ahmad Baktash and Mursal Dawodi. Gpt-4: A review on advancements and opportunities in natural language processing. *arXiv preprint arXiv:2305.03195*, 2023.
4. Pasquale Cantiello, Beniamino Di Martino, Michele Mastroianni, Luigi Colucci Cante, and Mariangela Graziano. Towards a cloud model choice evaluation: comparison between cost/features and ontology-based analysis. *International Journal of Grid and Utility Computing*, 2022.
5. Beniamino Di Martino, Luigi Colucci Cante, Antonio Esposito, and Mariangela Graziano. Towards a methodology for the semantic representation of iot sensors and bpmns to discover business process patterns: A smart irrigation case study. In Leonard Barolli, editor, *Advances on Broad-Band Wireless Computing, Communication and Applications*, pages 248–258, Cham, 2023. Springer International Publishing.
6. Beniamino Di Martino, Luigi Colucci Cante, and Nicla Cerullo. Opportunities and advantages of cloud migration of a smart restaurant system. In Leonard Barolli, Farookh Hussain, and Tomoya Enokido, editors, *Advanced Information Networking and Applications*, pages 153–162, Cham, 2022. Springer International Publishing.
7. Beniamino Di Martino, Mariangela Graziano, Luigi Colucci Cante, and Datiana Cascone. Analysis of techniques for mapping convolutional neural networks onto cloud edge architectures using splitfed learning method. In Leonard Barolli, Farookh Hussain, and Tomoya Enokido, editors, *Advanced Information Networking and Applications*, pages 163–172, Cham, 2022. Springer International Publishing.
8. Sijing Duan, Dan Wang, Ju Ren, Feng Lyu, Ye Zhang, Huaqing Wu, and Xuemin Shen. Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. *IEEE Communications Surveys & Tutorials*, 25(1):591–624, 2022.
9. Raul Castro Fernandez, Essam Mansour, Abdulhakim A Qahtan, Ahmed Elmagarmid, Ihab Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. Seeping semantics: Linking datasets using word embeddings for data discovery. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 989–1000. IEEE, 2018.

10. Reinaldo Padilha França, Rodrigo Bonacin, and Ana Carolina Borges Monteiro. An overview of data privacy and security in cloud platforms: Exploring the intersection of digital technologies currently. *Digital Cultural Heritage*, pages 21–39, 2024.
11. Othmane Friha, Mohamed Amine Ferrag, Burak Kantarci, Burak Cakmak, Arda Ozgun, and Nassira Ghoualmi-Zine. Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness. *IEEE Open Journal of the Communications Society*, 2024.
12. Yaqi Hu, Dongdong Ye, Jiawen Kang, Maoqiang Wu, and Rong Yu. A cloud-edge collaborative architecture for multimodal llms-based advanced driver assistance systems in iot networks. *IEEE Internet of Things Journal*, 2024.
13. Raisa Islam and Imtiaz Ahmed. Gemini-the most powerful llm: Myth or truth. In *2024 5th Information Communication Technologies Conference (ICTC)*, pages 303–308, 2024.
14. Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
15. Hongpeng Jin and Yanzhao Wu. Ce-collm: Efficient and adaptive large language models through cloud-edge collaboration. *arXiv preprint arXiv:2411.02829*, 2024.
16. Stefanos Laskaridis, Kleomenis Katevas, Lorenzo Minto, and Hamed Haddadi. Melting point: Mobile evaluation of language transformers. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 890–907, 2024.
17. Jieh-Sheng Lee. Lexgpt 0.1: pre-trained gpt-j models with pile of law. *arXiv preprint arXiv:2306.05431*, 2023.
18. Hanzhe Li, Sherry X Wang, Fu Shang, Kaiyi Niu, and Runze Song. Applications of large language models in cloud computing: An empirical study using real-world data. *International Journal of Innovative Research in Computer Science & Technology*, 12(4):59–69, 2024.
19. Guanqiao Qu, Qiyuan Chen, Wei Wei, Zheng Lin, Xianhao Chen, and Kaibin Huang. Mobile edge intelligence for large language models: A contemporary survey. *arXiv preprint arXiv:2407.18921*, 2024.
20. Kunal Rao, Giuseppe Coviello, Priscilla Benedetti, Ciro Giuseppe De Vita, Gennaro Mellone, and Srimat Chakradhar. Eco-llm: Llm-based edge cloud optimization. In *Proceedings of the 2024 Workshop on AI For Systems*, pages 7–12, 2024.
21. V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
22. Paulo Souza, Tiago Ferreto, and Rodrigo Calheiros. Maintenance operations on cloud, edge, and iot environments: Taxonomy, survey, and research challenges. *ACM Computing Surveys*, 56(10):1–38, 2024.
23. Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
24. Bing Wang, Shiyu Wang, and Qian Ouyang. Probabilistic inference layer integration in mistral llm for accurate information retrieval. 2024.
25. Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*, 2024.
26. Fan Yang, Zehao Wang, Haoyu Zhang, Zhenhua Zhu, Xinhao Yang, Guohao Dai, and Yu Wang. Efficient deployment of large language model across cloud-device systems. In *2024 IEEE 37th International System-on-Chip Conference (SOCC)*, pages 1–6. IEEE, 2024.
27. Zhengqing Yuan, Zhaoxu Li, Weiran Huang, Yanfang Ye, and Lichao Sun. Tinygpt-v: Efficient multimodal large language model via small backbones. *arXiv preprint arXiv:2312.16862*, 2023.
28. Xueying Zhai, Yunfeng Peng, and Xiuping Guo. Edge-cloud collaboration for low-latency, low-carbon, and cost-efficient operations. *Computers and Electrical Engineering*, 120:109758, 2024.
29. Mingjin Zhang, Xiaoming Shen, Jiannong Cao, Zeyang Cui, and Shan Jiang. Edgeshard: Efficient llm inference via collaborative edge computing. *IEEE Internet of Things Journal*, 2024.

30. Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.
31. Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. A review on edge large language models: Design, execution, and applications. *arXiv preprint arXiv:2410.11845*, 2024.