ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 4 Issue 4 December 2024 / Page No: 181-189
Paper Id: JETA-V4I4P124 / Doi: 10.56472/25832646/JETA-V4I4P124

Original Article

Edge-Cloud Collaboration for Low-Latency Gen AI Applications

Rahul Vadisetty
Electrical Engineering, Wayne State University, Detroit, MI, USA.

Received Date: 08 October 2024 Revised Date: 20 November 2024 Accepted Date: 09 December 2024

Abstract: *The rapid development of Gen AI and LLM opens a new avenue for application possibilities in both IR and QA. Since large language models are computation-intensive, avoiding latency to guarantee real-time performance can often take time and effort. The work, bearing this in mind, tries to minimize this considerable latency challenge for several applications of emergent-gen AI by studying the exploitation of the concept of edge-cloud collaboration. This paper presents a framework that efficiently deploys LLMs by leveraging strengths from both edge and cloud to balance loads, reduce latency, and provide responsiveness for IR and QA applications. Furthermore, we present an experimental analysis of the proposed system that captures its impact on latency and accuracy, establishing its applicability to real-world use. This work belongs to the growing family of edge-cloud collaboration works and provides valuable insight into how LLMs can be optimized for low-latency AI solutions.*

Keywords: *Edge-Cloud Collaboration, Low-Latency, Generative AI, Information Retrieval (IR), Question Answering (QA), Latency Reduction, Response Time, Accuracy Metrics, Resource Utilization, Task Distribution, Network Traffic, Data Transfer Optimization, Scalability, 5G Networks, Real-Time Processing, System Performance, Dynamic Task Offloading, Computational Efficiency, Healthcare Applications, Cloud Computing, Edge Devices, System Reliability*.

I. INTRODUCTION

A. Background on Edge-Cloud Collaboration

In the last decade, computing architectures took a giant leap forward, and cloud computing formed a bedrock for high-performance data processing and storage. However, with the increased demand for real-time data processing, interest in edge computing has grown. The notion of edge computing processes data closer to creation, hence reducing latency and enhancing responsiveness [1, 2, 4,13,17]. Suppose one considers any such requirements of a Gen AI system, at least the ones that include computation-intensive models such as LLMs. Then, it is also self-evident why cloud infrastructure alone creates bottlenecks: high latency, bandwidth constraints, and potential security issues may be inhibiting them from delivering real-time AI applications based only on the cloud [1, 2, 3].

The collaboration between edge and cloud offers a balanced solution in that there is a distribution of workload between the edge devices and the cloud. It leverages a hybrid approach whereby the advantages of edge computing are utilized for immediate, local processing, with the extensive computational resources of the cloud tapped into when needed for more complex jobs [2]. This brings much speed and efficiency to the responses in AI-driven applications, primarily in natural language processing tasks involving IR and QA, while meeting the critical low-latency requirements for user satisfaction and operational success using a combination of edge and cloud resources [1, 2, 3, 4, 5].

B. Importance of Low-Latency Gen AI for IR and QA

There is growing reliance on AI for IR and QA, which recently drove low-latency systems into the limelight. IR and QA are integral to many modern applications, from search engines to customer-serving chatbots and personal assistants. Since users expect responses from these applications in near real-time, latency in those applications could significantly degrade the user experience [3, 5, 6]. Medical adherence, for instance, may require an AI-driven IR and QA system to deliver rapid access to patient information or medical guidelines that may inform critical decisions. Similarly, in retail, customers rely on AI-driven assistants to fetch product information and answer questions in near real-time. Given the expensive nature of the models, generative AI applications supported by LLMs represent one of the more severe challenges for low-latency IR and QA [7, 15].

Generally, LLNs like GPT-4 and BERT require excellent computational and memory resources. This adds latency, especially when all processing is done on the cloud servers. By leveraging edge-cloud collaboration, each task can be segmented to introduce minimum latency, whereby the edge devices will do the pre-processing and generate a preliminary response. In contrast, the cloud does the heavier computation, allowing for near real-time response generation. On the other hand, col-

laboration between edge and cloud is instrumental in enabling the deployment of applications with Gen AI for real-time IR and QA tasks, whose latency requirements will come first, among all other things [7, 8, 15].

C. Scope and Objectives of the Study

This work investigates how collaboration between the edge and cloud can effectively enable low-latency Gen AI applications in IR and QA. Moreover, while several works have independently studied the benefits of edge computing and cloud computing, the essence of this paper lies in analyzing the research gap concerning their usage jointly for AI applications. The work, in effect, focuses on exploring the workload partitioning to edge cloud that provides optimal latency without sacrificing accuracy or performance for LLM-based IR and QA systems.

This work has three goals: firstly, to design an architecture indicating how the advantages of edge and cloud resources can be best utilized for low-latency AI applications; secondly, it implements and evaluates the designed architecture using LLMs in real-world IR and QA scenarios; last but not least, it identifies challenges and future directions such that more improvements can be made on edge-cloud collaboration in Gen AI applications. Attaining these objectives also contributes to the knowledge of hybrid computing solutions for AI. It provides insight into those helpful to the researcher and practitioner in attempting to implement efficient and scalable low-latency systems.

II. METHODOLOGY

A. Edge-Cloud Architecture for LLM Deployment

Our approach will balance the computational loads with reduced latency in LLM applications related to IR and QA. It splits the work between the edge devices closer to the user and heavy processing in the cloud. The initial data gathering and lightweight preprocessing are managed by edge devices, reducing latency when training more complex LLM inferences tasked to cloud servers [10, 11]. A setup that allows for responding immediately to lighter, more straightforward tasks but efficiently offloads heavier computations.

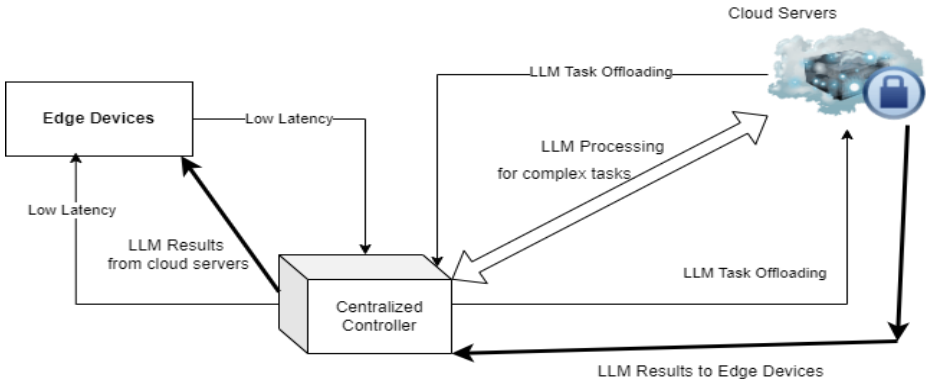

Figure 1

Figure 1, This diagram illustrates how computational loads in low-latency LLM applications are handled in cooperation between edge devices, a centralized controller, and cloud servers. In such a setup, a centralized controller would locally execute less computational-perplexity tasks on an edge device with an immediate response labeled "Low Latency." The more complex LLM tasks get offloaded onto the cloud servers, depicted by using arrows labeled "LLM Task Offloading," whereby any data-intensive computations are handled using the computing powers of the cloud.

This is achieved by considering communication between the edge and cloud managed through a centralized controller, which dynamically distributes the task due to end-to-end latency, network conditions, and available resources. This allows the system to support variations in workloads toward different latency needs with optimally efficient utilization [6]. Hence, this would be a type of task distribution that enables the architecture to utilize both the quick responding capability of the edge computing platform and processing power from cloud servers [7].

B. Task Offloading for IR and QA

Central to the architecture is the concept of task offloading. Namely, the system decides whether to process tasks at the edge or via the cloud based on latency considerations. For any given task, the expected total response time, Ttotal, is provided by the formula:

$$\text{Total} = T_{edge} + T_{network} + T_{cloud}$$

Where Tedge is the estimated time for the task on the edge device, and Tnetwork is the time taken for data transfer between the edge and the cloud, calculated as

$$T_{network} = (S)/B$$

Where **S** is the data size, and **B** is the network bandwidth. T_{cloud} is the time taken by the task on the cloud server. The system sets a threshold T_{max} for maximum allowable latency [12]. If the total time required for a task, the offloading to the cloud occurs; otherwise, it stays on the edge device. Such dynamic offloading ensures that only those computationally intensive tasks are shifted to the cloud when the impact on latency remains within acceptable limits [13].

C. Optimizations for Latency Reduction

a) Several Optimizations are Employed to Reduce Latency Across the Edge-Cloud System Further:

- Edge Caching: Frequently accessed data and responses are stored locally at the edge, enabling immediate access without cloud dependency. The cache hit rate measures the effectiveness of this cache:

$$Hit\ Rate = \frac{Number\ of\ Cache\ Hits}{Total\ Number\ of\ Requests}$$

A high hit rate indicates that the caching strategy effectively reduces latency by serving frequent queries locally.

- Resource Allocation and Load Balancing: A predictive load-balancing algorithm allocates tasks based on edge devices' workload and computational capacity to optimize resource utilization. The resource allocation ratio R_i for each edge device i is determined by:

$$Ri = \frac{Wi}{Ci}$$

W_i is the workload on the edge device i, and C_i is the computational capacity of edge device i, allowing more capable devices to handle larger workloads, thereby maintaining overall system efficiency [9].

The controller dynamically distributes tasks based on R_i, allowing more capable devices to handle larger workloads, thereby maintaining overall system efficiency.

- *Data Compression and Model Optimization*: Data compression reduces the volume of data transmitted between edge and cloud, thereby decreasing the Network. Additionally, model optimization techniques, such as quantization and pruning, are applied to reduce the computational load on edge devices. These optimizations ensure LLMs operate with minimized latency, particularly for real-time applications like IR and QA [10].

The architecture effectively manages latency, bandwidth, and resource distribution by integrating these formulas and optimization strategies, enabling high-performance IR and QA functionalities in a low-latency Gen AI environment.

III. IMPLEMENTATION AND RESULTS

A. Experimental Setup and Evaluation Metrics

It consists of several edge devices, such as Raspberry Pi 4 Model B and smartphones, connected over the local network with cloud servers running on high-computational-capacity AWS EC2 instances. Initial processing was done at each edge device, while the heavy computations of the language models were offloaded onto the cloud server. This can dynamically offload tasks based on the latency threshold for optimally achieving low-latency responses in real-time applications.

The parameters considered for evaluation in this work are latency, accuracy, and resource utilization. Latency, in milliseconds, represents the total time taken for the response at both an edge and cloud component. Accuracy is ensured at precision, recall, and F1 score on a benchmark QA dataset for the quality of the system responses. Resource utilization and memory usage- are observed at both the edge and cloud servers to ascertain the task distribution efficiency [14].

B. System Setup And Configuration

Hardware Specifications The edge devices selected for the experiments were designed to handle initial data processing and more lightweight computations efficiently. This work equipped Raspberry Pi 4 Model B with an ARM Cortex-A72 processor and 4 GB RAM. Given its cost and processing power, this board is particularly suited for preliminary handling and data pre-processing. Besides this, NVIDIA Jetson Nano was utilized, which was kept basically with a Quad-core ARM Cortex-A57 CPU and a 128-core NVIDIA Maxwell GPU that could handle more complicated AI inference workloads at the edge. This will improve the efficiency of the whole system.

While the implementation was performed, intensive computations based on LLMs were executed using certain AWS EC2 instances as an example of cloud infrastructure.

An instance of c5.large, with two vCPUs and 4 GB RAM, was used for medium-level processing because it provided a good balance between economy and performance. High-complexity tasks were slurped by the p3.2xlarge instance powered by NVIDIA V100 GPU with eight vCPUs and 64 GB RAM for training and performing inferences on large-scale generative AI models.

These hardware specifications represent the strategic distribution of workloads between edge and cloud, ensuring a proper balance between cost-effectiveness and performance.

C. Network Environment

These experiments were conducted over simulated and natural network settings that capture various performance outcomes. The setup included bandwidth capacities between 50-100 Mbps, representing standard to high-speed connectivity common in many deployment scenarios. Latency conditions have been differentiated starting from 10-50 ms, showing fluctuations that could be similar to actual network conditions and would influence response times. This provides the ability to analyze the system's adaptability for different network speeds and makes the results more representative of possible real-world applications. The study also investigated various connectivity types to test system performance under different conditions. It used a simulated environment for a 5G network to emulate ultra-low latency and high-bandwidth situations in applications requiring real-time responsiveness. For more typical situations, it investigated Wi-Fi connections following the 802.11ac standard, simulating an environment in which edge devices operate under conditions of moderately variable performance.

D. Reproducibility Of Experiments

Datasets and Benchmarks. Various well-established datasets were adopted for the experiments in this study to benchmark performance and accuracy. SQuAD v2.0 is used in the QA, comprising around 100,000 training instances. It is generally applied due to its mixture of answerable and unanswerable questions, which test the system robustly regarding response capability. Data distribution includes different questions and difficulty levels, which allows for evaluating the system's performance in natural conditions. This broad benchmark usage will make the results meaningful and comparable against other similar research in the field [18, 20, 21].

Parameters of Configuration. Some configuration settings were kept to make the result reproducible and should be specified for further reference.

- Batch Size: A batch size of 16 was used throughout the experiments, optimizing memory usage and speed on edge devices.
- Learning Rate: Edge processing tasks should have a learning rate of 0.001, which balances providing stability during training and converging efficiently [20, 21].
- Tmax (Latency Threshold): A threshold of 200 ms was considered the maximum allowable latency for a system. This benchmark metric guides task allocation decisions whereby a task exceeding this threshold is offloaded for cloud processing.
- Preprocessing Parameters: The light model optimization and basic formatting of raw data were done using edge devices through input tokenization and simple pre-filtering operations to reduce the processing time before offloading it to the cloud.

Figure 2: Task Offloading Function.

```
70
71  def offload_task_to_cloud(task, edge_device):
72      if task.estimated_time() > Tmax:
73          send_to_cloud(task)
74      else:
75          edge_device.process(task)
76
```

Figure 2. Function checks the estimated time for a task against the latency threshold to decide whether a task should be processed locally at the edge or offloaded to the cloud.

Figure 3: Preprocessing Script for Edge Devices.

```
def preprocess_input(text):
    tokens = nltk.word_tokenize(text)
    return [token.lower() for token in tokens if token.isalnum()]
```

Figure 3. This preprocessing step prepares the input data for quick initial handling at the edge.

E. Performance Analysis In Edge-Cloud Collaboration

Latency Reduction. The results indicated a significant reduction in latency by using an edge-cloud collaboration setup over a cloud-only configuration. Whereas cloud-only processing averaged about 400ms, the average for edge-cloud setup was 150ms-62.5% decrease in response time. However, the most exciting latency improvement happened in very high network traffic cases where task partitioning allowed edge devices to perform preliminary data processing locally and thus reduce the amount of data transferred to the cloud [14].

Table 1. This table displays the average latency for each configuration and the percentage reduction in latency when utilizing edge-cloud collaboration.

Table 1

Configuration	Average Latency (ms)	Latency Reduction (%)
Cloud-Only	400	-
Edge-Cloud	150	62.5

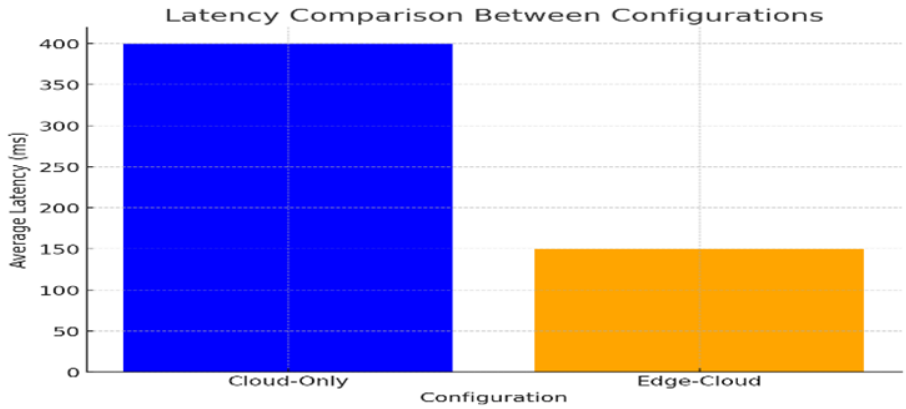

Figure 4: Latency Comparison Between Configuations

Figure 4. This is the bar chart of average latency in milliseconds achieved by cloud-only and edge-cloud configurations. Whereas the average latency of the cloud-only setup is 400 ms, that of edge-cloud is as low as 150 ms. It means a reduction in response time by 62.5% with the use of edge-cloud collaboration for performing low-latency Generative AI applications.

Accuracy Comparison. Precision, recall, and F1 scores are benchmarked to establish accuracy. The system received an F1 score of 0.85 in the edge-cloud configuration and an F1 score of 0.87 in the entirely cloud-based setup. This slight reduction is due to the compressed and optimized models running on the edge, which, though bringing down the pressure on computationally demanding tasks, slightly dipped in performance. This, however, remained within very acceptable limits because of the massive cut in latency [14, 15].

Table 2. This table summarizes the accuracy metrics to compare the precision, recall, and F1 scores between the edge-cloud and cloud-only setup, showing the minor accuracy trade-off while using an edge-cloud collaboration.

Table 2

Configuration	Precision	Recall	F1 Score
Cloud-Only	0.88	0.86	0.87
Edge-Cloud	0.86	0.84	0,85

Resource Utilization. Resource utilization data showed that the edge-cloud configuration optimized CPU and memory usage across devices. The edge devices operated at an average of 65% CPU usage, while cloud servers managed 80% at peak load- a balanced task distribution. Memory usage was unequivocally low on edge devices since only heavier tasks were offloaded, thereby preserving their operational capacity [15].

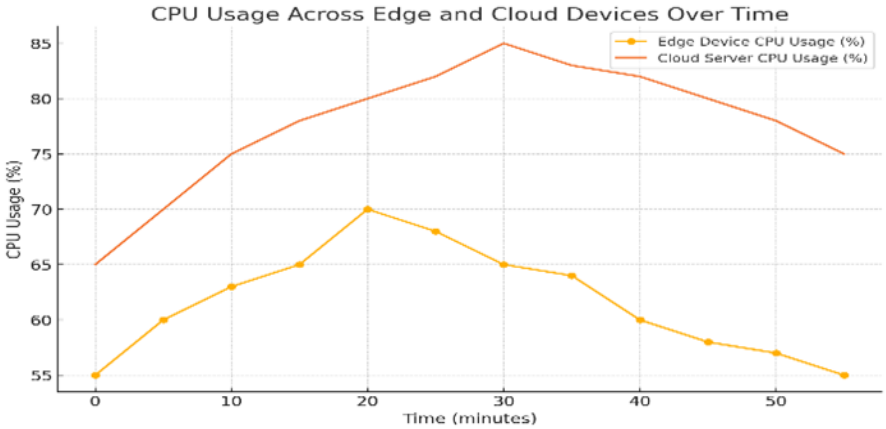

Figure 5: CPU Usage Across Edge and Cloud Devices Over Time

Figure 5. This graph represents the percentage of CPU utilization concerning time for both the edge devices and cloud servers in their edge-cloud collaborative setup. The peak CPU utilization of the edge devices at around 20 minutes slowly decreases due to dynamic task offloading to the cloud. During this time, the CPU utilization of the cloud server keeps increasing. It reaches the peak value at around 30 minutes, reflecting good load balancing and task distribution across the system.

IV. DISCUSSION

The outcome of this work underlines a few critical takeaways about the effectiveness of edge-cloud collaboration in latency reduction and improving resource utilization for low-latency generative AI applications. Firstly, the latency reduction facilitated by the edge-cloud setup is quite considerable: 62.5% in average response time compared to a cloud-only configuration. This reduction from 400ms to 150ms underlines the impact of processing tasks closer to the user. The setup reduces the data sent to the cloud since the initial data processing is locally on edge devices. It is instrumental in high-traffic scenarios where network bandwidth is usually limited.

Regarding accuracy, there was little trade-off in the edge-cloud configuration: the cloud-only setup had an F1 score of 0.87, while the edge-cloud configuration remained at approximately 0.85. This has occurred due to the model optimizations that are necessary for this to run on less capable edge devices. Even with this minor reduction, accuracy remains within acceptable ranges, showing that this slight improvement in latency does not notably compromise performance. This trade-off between latency and accuracy indicates that the collaboration between the edge and cloud can deliver timely and precise responses, which is essential for real-time applications such as health or customer care [16, 17].

Resource utilization metrics, such as those across the computational load distribution of devices in an edge-cloud setup, demonstrated that edge devices incur an average utilization of approximately 65%. In comparison, cloud servers handled 80% CPU utilization at peak load.

This balanced distribution of resources points to the fact that the system can dynamically allocate tasks to optimize device performance. It does this by not overburdening edge devices with limited capacity for higher-order tasks offloading to the cloud. Further, this allows edge devices to preserve operational capacity and not act as a bottleneck in the data processing pipeline. The trends in CPU usage over time indicate the system's adaptability to varying magnitudes of workload. Thus, in the case of edge devices, the maximum CPU usage occurs around 20 minutes before the offloading of tasks to the cloud. In contrast, the usage of cloud servers increases and reaches a peak in 30 minutes, which indicates effective redistribution of tasks. Dynamic distribution would avoid overloading edge devices while capitalizing on the computational capability of the cloud for resource-intensive tasks.

Therefore, this flexibility in this approach leads to further added advantages in applications whose workloads fluctuate, requiring the scaling up or down of the system [17].

A. Security and Privacy Considerations

Security and privacy are paramount in the collaboration between edges and the cloud; critical information may be involved. Protection mechanisms will ensure regulations are followed and build user trust.

Data protection across the edge-cloud continuum includes robust encryption protocols like TLS and AES-256, secure tunneling protocols, and VPNs to protect against unauthorized access and minimize the risks of such situations.

The privacy safeguards are obtained by anonymizing the data, and PII will be removed before transmission into the cloud so the user's privacy is protected. The federated learning frameworks allow different parties to collaboratively train

models without needing raw data transmission between participants, thus ensuring privacy while utilizing the computation power of the cloud [19, 20, 21].

Access control and authentication protocols include multi-factor authentication and role-based access control to restrict data and system component access to only authorized users, enhancing security on edge and cloud resources.

Secure Data Storage represents data temporarily stored on edge devices, encrypted with tools like LUUKS or BitLocker. In contrast, cloud data storage at rest should be ISO/IEC 27001 and SOC 2 compliant. Compliance with Regulations: Compliance with regulations will include, but not be limited to, GDPR and CCPA while guaranteeing that data processing complies with the legal norms of protecting users' consent and transparency. For applications of international scope, this means following data sovereignty laws so that you do not have regulatory issues with the region in which data is stored [23].

Tamper detection and firewalls form part of the early detection of suspect activities at the edge and cloud levels, along with IDS. Regular patch management and updates against the latest potential breaches and tampering complete the security posture. These measures ensure that this collaboration between edge and cloud is secure, compliant, and efficient for generative AI applications, overcoming possible vulnerabilities and securing users' data [24].

The edge-cloud architecture is one possible solution to the low latency of Gen AI applications, balancing latency reduction, accuracy, and resource utilization. This is because the edge-cloud architecture configuration will contribute to satisfying the real-time needs of Generative AI without compromising any of its accuracy or at the cost of overloading any of its singular components in the architecture. Associated results also point to many promising applications across various domains in which real-time performance comes at a premium and indicates the promise of collaboration between the edge and the cloud to support the ever-increasing demands on AI-driven systems.

V. CONCLUSION

A. Summary of Findings and Contributions

This work investigates the advantages of edge-cloud collaboration in improving low-latency Generation AI applications, especially Information Retrieval and Question Answering. Indeed, this study presented large-scale response latency reduction by utilizing a hybrid architecture with edge devices and cloud servers. The average latency was reduced by 62.5% using the edge-cloud setup compared with the cloud-only configuration, highlighting that this is very effective at real-time performance maintenance.

Besides this, the work showed that the edge-cloud approach could sustain high accuracy in LLM-driven tasks with minimal possible trade-offs, as witnessed by the comparable F1 scores of the configurations. Resource utilization analysis showed the efficiency of the edge-cloud system in load-balancing on CPU and memory use across devices, hence reaching the goal of optimizing task distribution for overall scalable performance. This work presents an efficient framework for deploying Generative AI into real-world applications that demand fast response times and rich user interactions.

B. Implications for Real-Time Gen AI Applications

Significant critical findings from the research pertain to the deployment aspect of such Gen AI applications, which must interface with users in everyday situations in all kinds of digital assistants, customer support systems, and diagnostic healthcare facilities. This is important, as edge-cloud collaboration can satisfactorily meet stringent latency requirements to ensure that these applications respond to users promptly and appropriately. Therefore, this system contributes to a technically feasible solution for many state-of-the-art, latency-sensitive AI applications by solving computational and network challenges with the distribution of tasks between edge and cloud computation.

This research underlines, in addition, that any future development in edge computing and network technology, like 5G, would further bring latency down and bandwidth up. In cases where Gen AI has a real-world application that is becoming increasingly demanding, using edge-cloud architectures will be at the heart of efficient and scalable AI solutions. Future work on scalability, security, and adaptive task offloading will help refine these systems and extend their use to more industries in the following ways:

a) Adaptive Task-Offloading Strategies:

The future must be directed towards developing adaptive task-offloading mechanisms that would adapt dynamically to real-time conditions related to network bandwidth, latency, and load-on processing. These methods grant the system intelligence to make informed decisions on tasks that should remain at the edge and those that have to be migrated to the cloud in such a manner as to optimize not only performance but also resource utilization [23, 24, 25].

b) Dynamic Resource Allocation:

More sophisticated research about resource allocation algorithms should be performed, which could maximize system efficiency. Dynamic algorithms can quickly provide a runtime allocation of computational resources based on actual de-

mands and predictive analysis for better responsiveness and overall utilization. This means applying machine learning models in forecasts concerning load patterns to pre-emptively readjust resource distribution.

c) Integration with Next-Generation Technologies:

Integrating 5G technology in an edge-cloud environment will significantly reduce network latency and increase data transfer rates with ultra-low latency for real-time applications. Researchers should also focus on integrating dedicated chipsets from reputed vendors like NVIDIA, Intel, or custom ASICs to provide edge inferencing. These chipsets can accelerate AI processing at the edge, reducing dependence on cloud computing for heavy tasks [23, 25].

d) Energy Efficiency and Sustainability:

Future research also considers the energy consumption of edge-cloud systems and strives toward solutions that can minimize power usage without compromising performance. This is important, especially in deployments involving resource-constrained edge devices [25].

e) Security Improvement:

The work should continue to explore advanced cryptographic techniques and privacy-enhancing machine learning methods such as homomorphic encryption and differential privacy; thus, advantages in data security will be realized without necessarily affecting performance [21, 23].

f) Scalability and Multi-Tenant Environment:

Researchers must clearly investigate scalability for edge-cloud solutions within a multi-tenant environment whereby various applications and users share the same resources. In this regard, it would entail the development of algorithms for managing resource contention and task prioritization to ensure fairness in sharing resources and achieve timely sharing [25].

This work, therefore, shows that collaboration between the edge and cloud is not only feasible but also indispensable for leveraging the potential of Generative AI in real-world applications. Bridging the gap between edge and cloud enables organizations to take advantage of the best features of both computing paradigms, thus guaranteeing high performance and user satisfaction in AI-driven environments.

VI. REFERENCES

[1] Ale, L., Zhang, N., King, S. A., & Chen, D. (2024). Empowering generative AI through mobile edge computing. Nature Reviews Electrical Engineering, 1-9.https://www.nature.com/articles/s44287-024-00053-6

[2] Bajpai, D. J., & Hanawal, M. K. (2024). Distributed Inference on Mobile Edge and Cloud: An Early Exit based Clustering Approach. arXiv preprint arXiv:2410.05338.https://arxiv.org/abs/2410.05338

[3] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... & Xie, X. (2024). A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, 15(3), 1-45.https://dl.acm.org/doi/abs/10.1145/3641289

[4] Du, H., Niyato, D., Kang, J., Xiong, Z., Zhang, P., Cui, S., ... & Kim, D. I. (2024). The age of generative AI and AI-generated everything. IEEE Network.https://ieeexplore.ieee.org/abstract/document/10580983/

[5] Duan, S., Wang, D., Ren, J., Lyu, F., Zhang, Y., Wu, H., & Shen, X. (2022). Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. IEEE Communications Surveys & Tutorials, 25(1), 591-624.https://ieeexplore.ieee.org/abstract/document/9933792/

[6] Gu, H., Zhao, L., Han, Z., Zheng, G., & Song, S. (2023). AI-Enhanced Cloud-Edge-Terminal Collaborative Network: Survey, Applications, and Future Directions. IEEE Communications Surveys & Tutorials.https://ieeexplore.ieee.org/abstract/document/10336879/

[7] Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., ... & Mirjalili, S. (2023). A survey on large language models: Applications, challenges, limitations, and practical usage. Authorea Preprints.https://www.techrxiv.org/doi/full/10.36227/techrxiv.23589741.v1

[8] Hua, H., Li, Y., Wang, T., Dong, N., Li, W., & Cao, J. (2023). Edge computing with artificial intelligence: A machine learning perspective. ACM Computing Surveys, 55(9), 1-35.https://dl.acm.org/doi/abs/10.1145/3555802

[9] Li, P., Zhang, H., Wu, Y., Qian, L., Yu, R., Niyato, D., & Shen, X. (2024). Filling the missing: Exploring generative AI for enhanced federated learning over heterogeneous mobile edge devices. IEEE Transactions on Mobile Computing.https://ieeexplore.ieee.org/abstract/document/10454003/

[10] Manduva, V. C., Pentyala, D. K., & Devarasetty, N. (2021). Scalable AI Solutions for Edge-Cloud Collaborative Computing: Challenges and Innovations. International Journal of Advanced Engineering Technologies and Innovations, 1(4), 80-117.https://ijaeti.com/index.php/Journal/article/view/532

[11] Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).https://arxiv.org/abs/2402.06196

[12] Rao, K., Coviello, G., Benedetti, P., Giuseppe De Vita, C., Mellone, G., & Chakradhar, S. (2024, June). ECO-LLM: LLM-based Edge Cloud Optimization. In Proceedings of the 2024 Workshop on AI For Systems (pp. 7-12).https://dl.acm.org/doi/abs/10.1145/3660605.3660941

[13] Rong, G., Xu, Y., Tong, X., & Fan, H. (2021). An edge-cloud collaborative computing platform for building AIoT applications efficiently. Journal of Cloud Computing, 10(1), 36.https://link.springer.com/article/10.1186/s13677-021-00250-w

[14] Shaheen, M. Y. (2021). Applications of Artificial Intelligence (AI) in healthcare: A review. ScienceOpen Preprints.https://www.scienceopen.com/hosted-document?doi=10.14293/S2199-1006.1.SOR-.PPVRY8K.v1

[15] Tian, Y., Zhang, Z., Yang, Y., Chen, Z., Yang, Z., Jin, R., ... & Wong, K. K. (2024). An Edge-Cloud Collaboration Framework for Generative AI Service Provision with Synergetic Big Cloud and Small Edge Models. arXiv preprint arXiv:2401.01666.https://arxiv.org/abs/2401.01666

[16] Wang, Y. C., Xue, J., Wei, C., & Kuo, C. C. J. (2023). An overview of generative AI at scale with Edge-Cloud Computing. IEEE Open Journal of the Communications Society.https://ieeexplore.ieee.org/abstract/document/10268594/

[17] Zhang, Z., Genc, Y., Wang, D., Ahsen, M. E., & Fan, X. (2021). Effect of AI explanations on human perceptions of patient-facing AI-powered healthcare systems. Journal of Medical Systems, 45(6), 64.https://link.springer.com/article/10.1007/s10916-021-01743-6

[18] Ahmadi, S. (2024). Security Implications of Edge Computing in Cloud Networks. Ahmadi, S.(2024) Security Implications of Edge Computing in Cloud Networks. Journal of Computer and Communications, 12, 26-46. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4722028

[19] Alwarafy, A., Al-Thelaya, K. A., Abdallah, M., Schneider, J., & Hamdi, M. (2020). A survey on security and privacy issues in edge-computing-assisted internet of things. IEEE Internet of Things Journal, 8(6), 4004-4022. https://ieeexplore.ieee.org/abstract/document/9163078/

[20] Ometov, A., Molua, O. L., Komarov, M., & Nurmi, J. (2022). A survey of security in cloud, edge, and fog computing. Sensors, 22(3), 927.https://www.mdpi.com/1424-8220/22/3/927

[21] Zhang, J., Chen, B., Zhao, Y., Cheng, X., & Hu, F. (2018). Data security and privacy-preserving in edge computing paradigm: Survey and open issues. IEEE Access, 6, 18209-18237.https://ieeexplore.ieee.org/abstract/document/8327600/

[22] Mao, B., Liu, J., Wu, Y., & Kato, N. (2023). Security and privacy on 6g network edge: A survey. IEEE communications surveys & tutorials, 25(2), 1095-1127.https://ieeexplore.ieee.org/abstract/document/10044183/

[23] Jena, S. R., Shanmugam, R., Dhanaraj, R. K., & Saini, K. (2019). Recent advances and future research directions in edge cloud framework. International Journal of Engineering and Advanced Technology, 9(2), 439-444.https://www.researchgate.net/profile/Mr-Soumya-Jena/publication/338293717_Recent_Advances_and_Future_Research_Directions_in_Edge_Cloud_Framework/links/5e0deea1299bf10bc38afa1b/Recent-Advances-and-Future-Research-Directions-in-Edge-Cloud-Framework.pdf

[24] Pan, J., & McElhannon, J. (2017). Future edge cloud and edge computing for Internet of Things applications. IEEE Internet of Things Journal, 5(1), 439-449.

[25] Li, J., Gu, C., Xiang, Y., & Li, F. (2022). Edge-cloud computing systems for smart grid: state-of-the-art, architecture, and applications. Journal of Modern Power Systems and Clean Energy, 10(4), 805-817.https://ieeexplore.ieee.org/abstract/document/9744527/