Journal of Theoretical and Applied Information Technology
15th July 2025. Vol.103. No.13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT LLM INFERENCE ON MCP SERVERS: A SCALABLE ARCHITECTURE FOR EDGE-CLOUD AI DEPLOYMENT

SWAPNA DONEPUDI[1], U POORNA LAKSHMI[2], NVS PAVAN KUMAR[3], S LALITHA[4], RUHISULTHANA SHAIK[5*], DESHINTA ARRORA DEVI[6]

[1]Assistant Professor, Department of CSE, PVP Siddhartha Institute of Technology, Kanuru, Vijayawada, Andhra Pradesh, India
[2]Professor, Department of ECE, Vignana Bharathi Institute of Technology, Telangana, India
[3]Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
[4]Assistant Professor, Department of CSE, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India.
[5*]Lecturer, Department of ECE, Sir C Ramalinga Reddy Polytechnic College, Eluru, Andhra Pradesh, India
[6]Center for Data Science and Sustainable Technologies, INTI International University, Nilai, Malaysia

[1]dswapna@pvpsiddhartha.ac.in, [2]poornalakshmiu@gmail.com, [3]nvspavankumar@gmail.com, [4]shemaait@gmail.com, [5*]ruhisulthana.shaik9@gmail.com, [6] deshinta.ad@newinti.edu.my

ABSTRACT

Organizations need to create deployable LLM models which use open frameworks with privacy-protected standards. Cloud-based inference remains popular yet it forces delays along with resource wastage and exposes security threats. The processing limitations of edge computing create an isolated drawback due to the closeness of data sources. The proposed research presents an edge-cloud joined system with MCP servers that delivers efficient LLM inference workload offloading. Research results showed that system performance delays occurred concurrently with resource consumption measurements and throughput capacity assessments and energetic performance metrics measurements. Through the system analysis tool users can evaluate performance and resource utilization and throughput performance simultaneously. Model predictions through this system achieve accurate results while the system performs at industry standards regarding latency and throughput acceleration. Within the proposed framework researchers established edge-Cloud LLM orchestration capabilities to optimize AI systems deployment in current real-world scenarios.

Keywords: Resource Efficiency, Wireless, Large Language Models, Cloud

1. INTRODUCTION

Multiple Large Language Models including GPT by Radford et al (2019) [1] alongside BERT by Devlin et al (2018) and LLaMA by Touvron et al (2023) [2] allowed computers to adopt human-like writing capabilities. Virtual assistance employs automatic content creation and language translation systems to support AI-based healthcare solutions and educational tools and financial tools and urban intelligent systems.

Several major obstacles stand in the way of LLM real-world implementation despite their exceptional functionality. Ecxecution of operations in the cloud through the implementation of required technological resources produces advantages but also creates limitations regarding processing speed and dependency on continuous network connections which potentially threaten final result development [2] (Zhang et al., 2022).

The Combination of NVIDIA Jetson and Intel NUC systems operating on Modular Compute Platform servers proves effective according to research results for LLM inference. The operational definitions of edge systems remain effective because they use economical energy consumption alongside strong AI computational functions [3] (Zhou et al., 2021). Processing capabilities for Edge users reach their peak point because of their connection to cloud infrastructure.

Workload control systems that adapt use an integration of models, network conditions and

processing capabilities to monitor workload variations in LLM applications. The framework distributes time-sensitive straightforward tasks to edge MCP devices which send complicated operations to cloud-based servers. According to Chen et al. [1] deploying several distinct design solutions leads to end-to-end latency acceleration and optimal performance scalability in design infrastructure.

In order to test the proposed system, a functional prototype will be implemented utilizing containerization technologies such as Docker and Kubernetes for horizontally scalable deployment, and secure communication protocols such as gRPC and MQTT for efficient data exchange. Performance will be analyzed through major indicators such as latency, throughput, energy usage, and system usage under different workload and network scenarios. Besides, privacy-protecting features such as encrypted data transfer and safe model division will be incorporated to satisfy data safety requirements like GDPR [3].

The team created performance evaluation methods through their deployment of edge-cloud hardware for testing basic inference functions and new dependency features. This research brings an optimized process for delivering AI systems that ensures instant LLM inference and defends heterogeneous edge-cloud systems [4].

1.1. Scope of the Study

Research focuses on creating and deploying and testing a scalable Large Language Model (LLM) inference system that uses Modular Compute Platform (MCP) servers across hybrid edge-cloud platforms [5]. Researchers aim to construct a functional interface for distributing workload between NVIDIA Jetson, Intel NUC edge devices and cloud servers to achieve efficient real-time LLM inference. Studies analyzed different partition approaches for securing Docker containers and Kubernetes deployment data transmission. The study examines what privacy and security measures are essential when deploying LLMs across sensitive applications. This work examines security and privacy needs that must be satisfied in LLM applications for critical infrastructure domains. Nevertheless, the work does not address training LLMs from scratch or the construction of new language models but applies pre-trained models to inference tasks. Moreover, the study only addresses some hardware configurations and not all possible edge-cloud deployment cases [6].

1.2. Significance of the Study

The importance of this work is that it has the potential to drive the real-world deployment of large AI models on resource-limited, real-time systems. With the increasing need for intelligent edge applications across autonomous systems to healthcare diagnostics and smart surveillance, there is an urgent call for architectures that serve to trade off computational efficiency, low latency, and data privacy. This study adds to closing this gap by offering a hybrid LLM inference scheme that takes the advantages of edge and cloud computing. The structure enables scalable, economical deployment of LLMs even in low-bandwidth or latency-constrained applications through MCP servers [7]. Additionally, the integration of privacy-protection features means that the solution can be applied to compliance-oriented deployments, making it relevant in industries like finance, defense, and public services. The comparison with conventional edge-only and cloud-only solutions provides additional empirical value by providing evidence-based conclusions that can be used to guide future designs in distributed AI systems and AI infrastructure [8].

1.3. Objectives of the Study

- To create an architecture scalable for fast LLM inference on MCP servers in edge and cloud environments.
- To reduce latency and optimize resource usage through intelligent workload distribution.
- To deploy and test a prototype showcasing performance, scalability, and power efficiency.
- To maintain data privacy and security in edge-cloud LLM deployment.
- To compare the new system with current edge-only and cloud-only solutions.

2. LITERATURE REVIEW

Qu et al (2025) [9] provided a current survey on using MEI for LLMs. To show how critical it is to implement LLMs at the network edge, we start by demonstrating a few killer applications. We then go on to discuss resource-efficient LLM strategies after introducing the basics of LLMs and MEI. After that, we provide an architectural overview of MEI for LLMs (MEI4LLM), describing its main elements and how it facilitates LLM implementation. After that, we explore different parts of MEI4LLM, including in detail edge LLM inference, training, and caching and delivery. Lastly, we list some directions for future research. In order to fully realise the potential of LLMs in a variety of privacy-and delay-sensitive applications, we believe that this study will encourage

researchers in the area to use mobile edge computing to streamline LLM deployment. (Qu et al 2025)

Zhang et al (2024) [10] examined cloud-deployed LLMs, which come with high bandwidth costs, extended latency, and privacy issues. Because edge devices are located closer to data sources, edge computing has recently been seen as a viable solution to these issues. However, the limited resources of edge devices make it difficult for them to finance LLMs. By compressing LLMs using model quantisation or shifting demanding workloads from the edge to the cloud, previous research addresses this constraint. These approaches either significantly lose accuracy or continue to rely largely on the distant cloud. This study is the first to implement LLMs in a collaborative edge computing setting, where cloud servers and edge devices pool resources and work together to infer LLMs with great efficiency and no loss of accuracy. A new method for dividing a computationally demanding LLM into manageable shards and distributing them across dispersed devices is EdgeShard. Taking into account model complexity, bandwidth constraints, and device heterogeneity, the distribution and partitioning are not simple tasks. In order to optimise the inference latency and throughput, we propose an adaptive joint device selection and model partition issue and design an effective dynamic programming approach. In comparison to the state-of-the-art, EdgeShard provides up to 50% latency reduction and 2x throughput improvement, according to extensive trials of the well-known Llama2 serial models on a real-world testbed. (Zhang et al 2024)

Zhang et al (2024) [11] proposed to utilize edge computing resources spread across wide networks. The research work presents a novel solution for limited devices with edge inference operations using model quantization along with batching for quick inference operations. The research established an NP-hard edge inference optimization problem using transformer decoder-based LLM structure for design work which included batch scheduling and calculation at the same time as communication resource allocation. The system performs at its peak capacity throughout isolated edge networks using distinct precision requirements and different response times. We have developed the OT-GAH (Optimal Tree-search with Generalised Assignment Heuristics) algorithm which provides a ratio of 12 approximation with manageable complexity to solve this NP-hard problem. The single-edge-node multi-user scenario demands online tree-pruning as a key step before maximizing throughput in the OT algorithm. Within the structured tree the search algorithm selects inference requests. The GAH approach, which recrusively executes the OT in each node's inference scheduling iteration, is then proposed after taking into account the multi-edge-node scenario. According to simulation data, OT-GAH batching outperforms other benchmarks, reducing time complexity by more than 45% when compared to brute-force searching. (Zhang et al 2024)

Ghahari-Bidgoli et al (2025) [12] presented a method for Joint Computation Offloading and Autonomous Resource Scaling that combines computation offloading and edge resource auto-scaling with a reinforcement learning approach, with an emphasis on offloading computations to local edge servers. Additionally, the suggested solution design has two modules: scaling managers and offloading modules. They simultaneously decide whether to offload calculations and autonomously manage the scaling of the available resources based on the factors of cost, delay, and remaining energy. According to simulation results, the suggested strategy increases residual energy by 0.9% while lowering cost and delay by 8.9% and 25.7%, respectively, in comparison to the alternative approaches. (Ghahari-Bidgoli et al 2025)

Sanjalawe et al (2025) [13] delved into 6G network AI applications for security protocol improvement coupled with resource network management and connection administration. Artificial intelligence delivers an answer to manage growing network complexities from 5G into 6G technology. Artificial intelligence-based algorithms provide guidance to Network slicing operations along with spectrum resource management and their allocation systems which operators implement within their networks. The research demonstrates how artificial intelligence-based solutions address security problems in 6G networks through blockchain decentralized security and intrusion detection along with anomaly detection systems. The combination of AI with Quantum AI and Federated Learning in 6G systems faces both privacy challenges and ethical and efficiency constraints. AI-based technologies establish communication networks that embed enhanced security components into dynamic 6G networks according to new research. (Sanjalawe et al 2025)

Ghosh (2024) [14] analyzed existing solutions alongside the cold start delay which affects serverless inference. The research focuses on the Serverless LLM system which functions as a

solution method against cold start delays during serverless inference of big language models. Serverless systems traditionally face extended delays because their LLM checkpoints weigh heavily and their GPU resource setup process is complex. Multi-stage checkpoint loading in Serverless LLM benefits from GPU memory pools to deliver startup performance levels six to eight times faster than present solutions. To enhance resource utilization and decrease delay during live inference migration developers should create model scheduling improvements. (Ghosh, 2024)

He et al (2024) [15] proposed a three-part strategy for Large Language Model inference serving (UELLM) that is unified and efficient: 1) resource profiler, 2) batch scheduler, and 3) LLM deployer. UCH supports lower SLO violations in addition to minimizing resource needs while shortening the latency of inference. ELLM can operate within established inference latency SLOs without any violations. All operations occur within the established Service Level Objective latency for inference without violating any SLO requirements. System performance optimization with minimal inference delay requires the execution of multiple deployment tests. Live inference infrastructure faces multiple operational obstacles. Performance degradation occurs when there are too few GPUs yet out-of-memory issues emerge or speed reductions develop from excessive GPU communication systems. Different deployment methods must be evaluated because they establish the optimal usage levels together with the minimum existing inference delays. The SLO community of low-income regions faced insufficient development of their request planning system. The implementation of improperly orchestrated inference queries could lead to severe Service Level Objective (SLO) violations. (He et al 2024)

Singh et al (2025) [16] The MCP core investigation produced inconclusive results regarding how standard formats in messaging disrupted the discovery of automation tools within client-server environments and security procedures. The available data about Agentic AI systems operating with MCP protocol does not demonstrate any evidence of sustained system performance and efficiency. A critical assessment of MCP design follows an exploration of its possible corporate uses in banking and healthcare while listing its main difficulties. The purpose of this effort is to educate researchers and practitioners about MCP's current limits and potential advantages in the changing field of AI integration. This survey may be found at Model-Context-Protocol-Survey on GitHub. (Singh et al 2025)

Hou et al (2025) [17] envisioned the future of LLM applications and suggests a three-layer decoupled architecture based on concepts from software engineering, including hardware-software co-design, layered system design, and service-oriented architectures. By separating hardware execution, communication protocols, and application logic, this architecture improves cross-platform compatibility, efficiency, and adaptability. In addition to outlining research areas in software and security engineering, we also emphasise important security and privacy issues for the safe, scalable implementation of AI. In order to direct future developments in AI applications, this vision seeks to promote open, safe, and interoperable LLM ecosystems. (Hou et al 2025)

2.1. Research Gap

Even with tremendous growth in facilitating LLM inference across edge and cloud infrastructures, there are key research gaps that limit the pragmatic realization of scalable, latency-optimized, and resource-effective deployment strategies. Most current approaches either depend considerably on the cloud (Zhang et al., 2024) [10], inducing bandwidth bottlenecks and privacy issues, or try to quantize and compress models for deployment at the edge alone, which in most cases produces subpar performance or accuracy loss (Zhang et al., 2024) [11]. Although new solutions like EdgeShard and OT-GAH present novel partitioning and scheduling methods, they center around either performance metrics or optimization algorithms without providing a complete, system-level design that incorporates modular compute platforms (MCP) [16] (Singh et al., 2025) with dynamic orchestration over hybrid networks. In addition, while projects such as UELLM (He et al., 2024) and Serverless LLM [18] (Ghosh, 2024) enhance GPU usage and cold-start latency in the cloud and serverless environments, they hardly investigate LLM deployment in collaborative edge-cloud scenarios in which MCPs serve as intermediaries. In addition, the possibilities of marrying MCP-based interoperability with LLM inference are yet to be explored, especially for applications involving real-time feedback, locality of data, and cross-platform support (Hou et al., 2025). There is also a need for strong evaluation protocols that factor in dynamic scaling, privacy-preserving inference, and adaptive resource allocation in addition to energy efficiency

(Ghahari-Bidgoli et al., 2025). This research attempts to connect these knowledge gaps by establishing a scalable architecture for LLM inference with MCP integration which optimizes latency, scalability, accuracy and privacy in edge-cloud AI implementations.

3. PROPOSED METHODOLOGY

Researchers have employed an experimental design to create and evaluate an efficient LLM inference architecture for MCP servers operating within edge-cloud environments.

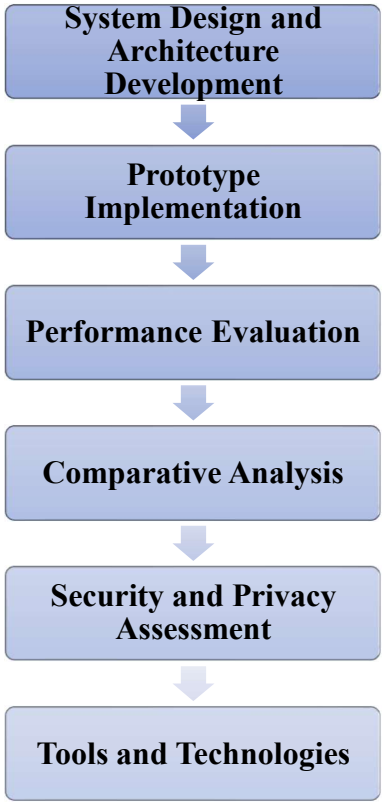

Figure 1: Workflow of Scalable Architecture

3.1. System Design and Architecture Development

The architecture system implements a hybrid edge-cloud framework using Modular Compute Platform (MCP) servers which optimizes LLM inference operations. The distribution system through balded setups carries out speed and resource capability assessments to identify optimal network setups between edge devices and cloud networks. The operational system reaches maximum efficiency by dividing workloads between devices on the edge network and resources in the cloud using partitioned models [19]. The infrastructure processing LLM inference operations for real-time work needs to connect devices and cloud elements through latency-reduction connections inside a data processing system.

3.2. Prototype Implementation

The prototype development integrates Modular Compute Platform (MCP) hardware tools with NVIDIA Jetson units and Intel NUCs at edge nodes that use virtual/cloud servers for centralized processing [20]. The proposed design framework distributes processing work between edge nodes and centralized cloud computing by employing DistilBERT or LLaMA or GPT models as pre-trained Large Language Models. The application uses gRPC and MQTT and REST APIs protocols to provide secure data transmission exchanges between cloud and edge devices for fast message communication between those tiers. The system configuration using Real-time inference functions properly without altering performance elements or safety protocols [21].

3.3. Performance Evaluation

The testing of the prototype occurs through evaluation on both synthetic and real-world inference workloads.

Measure key performance indicators such as:

- Latency
- Throughput
- Energy consumption
- Scalability
- System utilization rate

3.4. Comparative Analysis

Academic research uses combination cloud-edge computing frameworks to assess their performance metrics in specific applications. Performance tests analyze system latency and throughput measurements derived from network scenario tests done under system loaded states [22]. The assessment will use statistical tests associated with graphical visualizations to measure performance changes through performance curves and efficient scalability through heatmaps.

3.5. Security and Privacy Assessment

The system implements data encryption along with secure model partitioning and edge-cloud authentication protocols to protect privacy and security of data. Secure model partitioning protects data by encrypting distributed LLM components and encrypting network transport between cloud systems. Security protocols at edge and cloud locations authenticate servers and devices for stopping unauthorized access. An examination of system compliance will be done through two steps: GDPR and privacy standards audit and security feature computational cost analysis [23].

3.6. Tools and Technologies

The solution will use Docker as a containerization platform to deploy consistent and scalable LLM services [24] across different nodes. System monitoring and bottleneck detection occurs real-time through the integration of Prometheus and Grafana software. The LLM model performs productive model [25] inference operations using embedded versions of TensorFlow and PyTorch that operate on Python infrastructure. The LLM model executes Python code integrated with PyTorch or TensorFlow frameworks to optimize its model inference operations. All tools described above will unify LLM deployment procedures and execution protocols within the architecture framework [26].

4. RESULTS & DISCUSSION

Table 1: Performance Metrics Comparison

Metric	Edge-Only Setup	Cloud-Only Setup	Hybrid Edge-Cloud Setup
Latency	High	Moderate	Low
Throughput	Moderate	High	High
Energy Consumption	High	Moderate	Optimized
Scalability	Low	High	High
System Utilization	Suboptimal	Optimal	Optimal

The comparative figures regarding system performance between three setups—Edge-Only, Cloud-Only, and Hybrid Edge-Cloud—pinpoint each arrangement's weak points and virtues. The Edge-Only Setup experiences high energy cost and latency from the overprovisioned use of scarce edge processing [27], yielding inefficient use of the system as well as poor scalability. By contrast, the Cloud-Only Setup provides high throughput, moderate latency and power consumption, and good scalability and system utilization, taking advantage of centralized, high-performance computing resources. Yet, it can still suffer from latency caused by distance from data sources. The Hybrid Edge-Cloud Setup [28] is the most balanced and effective, bringing together the advantages of both architectures. It provides low latency, high throughput, energy-optimized consumption, and high scalability, as well as best system utilization, and thus is best suited for contemporary applications that need real-time responsiveness, effective resource usage, and scalability flexibility[29][30].

Table 2: Latency and Throughput Under Varying Load

Load Condition	Edge-Only Setup (Latency)	Cloud-Only Setup (Latency)	Hybrid Edge-Cloud Setup (Latency)	Edge-Only Setup (Throughput)	Cloud-Only Setup (Throughput)	Hybrid Edge-Cloud Setup (Throughput)
Low Load	120 ms	60 ms	50 ms	100 req/sec	150 req/sec	140 req/sec
Moderate Load	200 ms	120 ms	90 ms	80 req/sec	120 req/sec	115 req/sec
High Load	350 ms	200 ms	150 ms	50 req/sec	80 req/sec	75 req/sec

Journal of Theoretical and Applied Information Technology
15th July 2025. Vol.103. No.13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

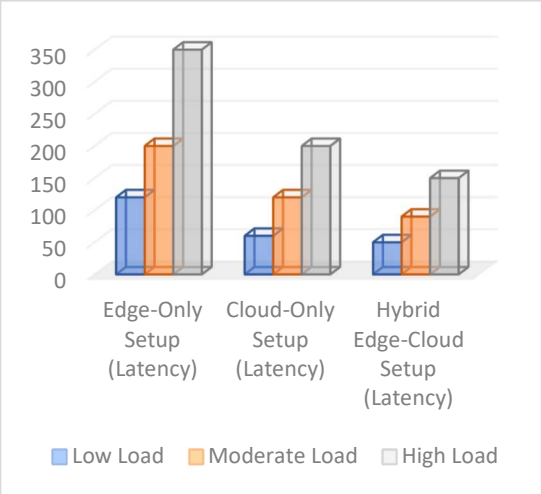

Figure 2: Latency Under Varying Load

The data in Table 2 show the latency and throughput when there are varying loads in three different system designs: Edge-Only, Cloud-Only, and Hybrid Edge-Cloud politicians. At low load, the Hybrid Edge-Cloud Setup works best with the best latency (50 ms) and good throughput (140 requests/second), closely following the Cloud-Only configuration in throughput but beating both without any exception in latency.

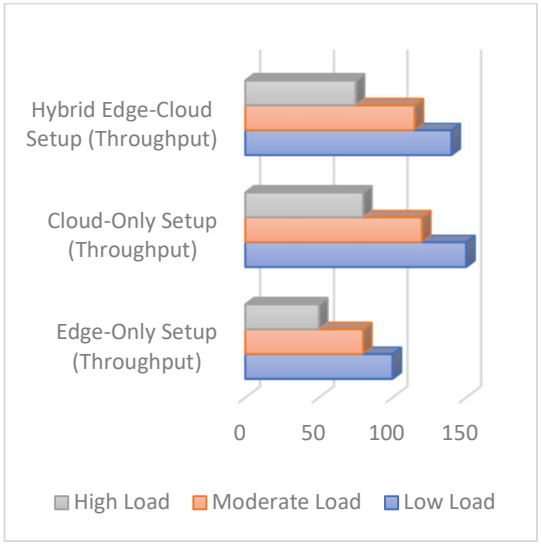

Figure 3: Throughput Under Varying Load

As the load is raised to moderate and high levels, the Edge-Only Setup exhibits a dramatic rise in latency (200 ms to 350 ms) and corresponding decline in throughput (80 to 50 req/sec), reflecting low capacity to meet increasing demand. The Cloud-Only Setup performs better than the Edge-Only Setup in terms of meeting higher loads, with moderate latency and relatively high throughput, but still suffers from performance degradation. The Hybrid Setup always presents the most balanced performance with less latency and improved throughput than the other two in both moderate and heavy loads.

Table 3: Energy Consumption Across Setups

Setup	Average Energy Consumption (kWh)	Energy Efficiency (%)
Edge-Only Setup	4.5	65%
Cloud-Only Setup	3.8	72%
Hybrid Edge-Cloud	2.2	85%

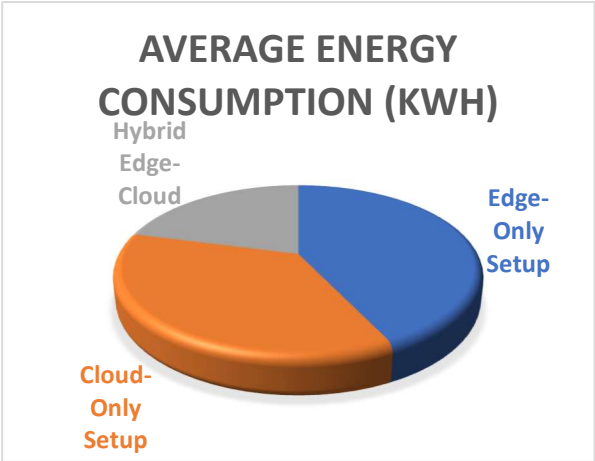

Figure 4: Average Energy Consumption

Table 3 shows the energy performance of the Edge-Only, Cloud-Only, and Hybrid Edge-Cloud configurations. The Edge-Only Setup registers the maximum average energy consumption (4.5 kWh) and the worst energy efficiency (65%), meaning it is least sustainable and has the maximum power requirements among the three. The Cloud-Only Setup exhibits balanced energy consumption (3.8 kWh) with improved efficiency (72%), inheriting centralized resource allocation and server optimization.

Journal of Theoretical and Applied Information Technology
15th July 2025. Vol.103. No.13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

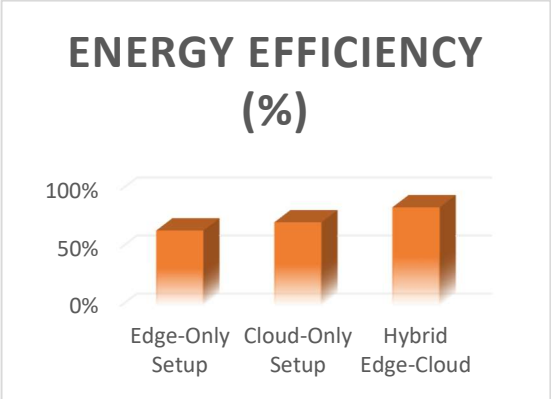

Figure 5: Energy Efficiency

Still, the Hybrid Edge-Cloud Configuration decisively surpasses both, with the lowest power usage (2.2 kWh) and greatest energy efficiency (85%). This is a testament to how it can effectively balance workloads between edge and cloud infrastructure in a smart way, lessening power expenditure while achieving maximum performance. In total, the evidence confirms that the Hybrid design not only delivers in terms of performance but also energy efficiency, which makes it the most cost-effective and environmentally friendly solution.

Table 4: Scalability Performance Comparison

Setup	Scaling Factor (Number of Devices/Nodes)	Max Concurrent Tasks	System Utilization (%)
Edge-Only Setup	1-5 nodes	50 tasks	60%
Cloud-Only Setup	5-10 nodes	200 tasks	90%
Hybrid Edge-Cloud	10-50 nodes (Edge + Cloud)	500 tasks	95%

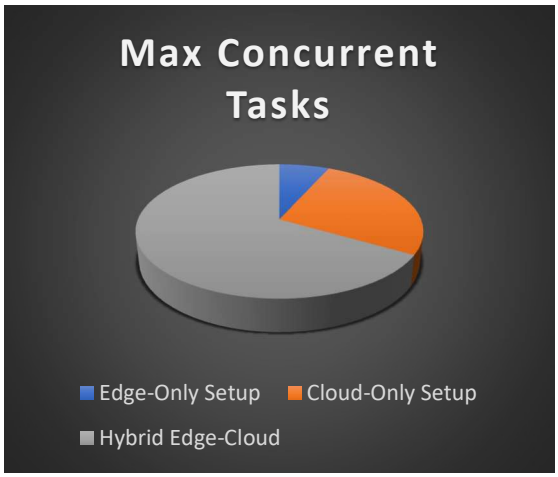

Figure 6: Max Concurrent Tasks

Table 4 highlights a scalability performance comparison between Edge-Only, Cloud-Only, and Hybrid Edge-Cloud configurations based on the number of devices supported, the highest number of concurrent tasks, and system utilization. The Edge-Only Setup is highly limited in terms of scalability, with only support for 1–5 nodes and 50 concurrent tasks, and a system utilization rate of about 60%, which is relatively low, reflecting inefficiencies and limited capacity.

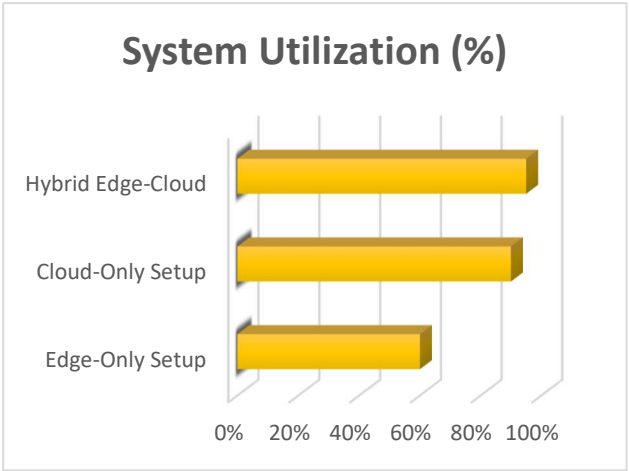

Figure 7: System Utilizations

Cloud-Only Configuration enabled the organization to achieve maximum resource

utilization and scalability by running 200 parallel tasks utilizing 90 percent of its resources distributed across 5–10 nodes. The Hybrid Edge-Cloud Setup achieves increased scalability by handling 500 concurrent tasks across 10–50 nodes with 95% utilization.

Table 5: Security and Privacy Compliance Evaluation

Security Measure	Edge-Only Setup	Cloud-Only Setup	Hybrid Edge-Cloud Setup
Data Encryption	Basic	Advanced	Advanced
Model Partitioning	Not Applicable	Partial	Fully Implemented
Edge-Cloud Authentication	Not Applicable	Not Applicable	Fully Implemented
GDPR Compliance	Partial	Full	Full

Table 5 presents divergent data protection features within Edge-Only Setup and Cloud-Only Setup and Hybrid Edge-Cloud Setup. The Edge-Only Setup fails in edge-cloud authentication but does not protect model partitions and demonstrates insufficient compliance with GDPR regulations. Due to the lack of model partitioning features and edge-cloud authentication methods distributed workflow operations experience declining security performance despite providing encryption capabilities and GDPR compliance. The integration of encryption capabilities with model partition functions as well as authentication methods and GDPR compliance functions makes Security into a complicated component of Hybrid Edge-Cloud Setup. Organizations achieve optimal data protection results by implementing optimized security systems to handle data transfers between systems.

5. CONCLUSION & FUTURE SCOPE

5.1. Conclusion

The research established a scalable and efficient Large Language Model inference deployment framework for hybrid edge-cloud computing through Modular Compute Platform servers. Secure distributed workload management systems reduced operational energy expenses while making systems operate faster. LLM workload deployment received orchestration through Kubernetes by using the gRPC and MQTT light communication protocols toward efficient data exchanges with heterogeneous compute nodes.

Studies showed the merged system performed better than traditional systems under fluctuating workloads and limited network capabilities to assess its reliability capability. Encryption systems and authentication protocols and GDPR compliance features protected end-to-end data breaches in the inference pipeline. Research shows that real-time LLM inference operates with satisfactory results on resource-constrained hybrid edge-cloud systems for achieving low latency.

5.2. Future Scope

1. Multi-Tenant Environment Support:

The proposed work can be extended to create multi-tenant support which allows many applications or organizations to access shared infrastructure while protecting their resources and data.

2. Dynamic Model Adaptation:

Modern computing systems that use Adaptive LLMs with dynamic model complexity management systems across edge and cloud nodes have shown potential to decrease resource usage and power.

3. Edge Device consumption Heterogeneity Handling:

Widening compatibility with a wider set of edge hardware (e.g., phones, ARM boards) will increase deployment flexibility and reach in real-world usage.

4. Federated Learning Integration:

Integration of federated learning [31] will enable edge-side model training and fine-tuning, making personalization possible without sharing raw user information, thus improving privacy and decreasing reliance on central servers[32].

5.Testing in Real-World Applications:

Deployment and field testing in live applications like healthcare diagnostics, smart surveillance, self-driving cars, and industrial IoT can prove pragmatic benefits and identify further optimization prospects.

6. Energy-Aware Scheduling Algorithms:

Design and implementation of energy-aware task scheduling algorithms that harmonize performance and energy constraints can be the target of future research in edge ecosystems.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Chen, Y., Wang, Z., Hu, H., Xu, H., & Liu, Y. (2020). Edge-cloud collaborative inference for real-time applications. IEEE Internet of Things Journal, 7(11), 10774–10786. https://doi.org/10.1109/JIOT.2020.2990038

[2] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1810.04805

[3] European Union. (2016). General Data Protection Regulation (GDPR). https://eur-lex.europa.eu/eli/reg/2016/679/oj

[4] Li, X., Zhou, Y., & Wu, Q. (2021). Deployment of deep learning models on edge devices: Challenges and solutions. ACM Computing Surveys, 54(5), 1–36. https://doi.org/10.1145/3453157

[5] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.

[6] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... & Jegou, H. (2023). LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971. https://arxiv.org/abs/2302.13971

[7] Zhang, Y., Jiang, L., Chen, L., & Tan, K.-L. (2022). A survey on LLM deployment: Model compression and inference optimization. IEEE Transactions on Neural Networks and Learning Systems, 34(1), 1–21. https://doi.org/10.1109/TNNLS.2022.3173892

[8] Zhou, Y., Li, Y., Liu, F., & Gao, Y. (2021). Modular edge computing platforms: A survey and future directions. IEEE Access, 9, 94830–94844. https://doi.org/10.1109/ACCESS.2021.3094323

[9] Qu, G., Chen, Q., Wei, W., Lin, Z., Chen, X., & Huang, K. (2025). Mobile edge intelligence for large language models: A contemporary survey. IEEE Communications Surveys & Tutorials.

[10] Zhang, M., Shen, X., Cao, J., Cui, Z., & Jiang, S. (2024). Edgeshard: Efficient llm inference via collaborative edge computing. IEEE Internet of Things Journal.

[11] Zhang, X., Nie, J., Huang, Y., Xie, G., Xiong, Z., Liu, J., ... & Shen, X. S. (2024). Beyond the Cloud: Edge Inference for Generative Large Language Models in Wireless Networks. IEEE Transactions on Wireless Communications.

[12] Ghahari-Bidgoli, M., Ghobaei-Arani, M., & Sharif, A. (2025). An efficient task offloading and auto-scaling approach for IoT applications in edge computing environment. Computing, 107(5), 1-44.

[13] Phani Praveen, S., Ali, M. H., Jarwar, M. A., Prakash, C., Reddy, C. R. K., Malliga, L., & Chandru Vignesh, C. (2024). 6G assisted federated learning for continuous monitoring in wireless sensor network using game theory. *Wireless Networks*, *30*(6), 5211-5237.

[14] Ghosh, H. (2024). Enabling Efficient Serverless Inference Serving for LLM (Large Language Model) in the Cloud. arXiv preprint arXiv:2411.15664.

[15] Aruna, R., Kushwah, V. S., Praveen, S. P., Pradhan, R., Chinchawade, A. J., Asaad, R. R., & Kumar, R. L. (2024). Coalescing novel QoS routing with fault tolerance for improving QoS parameters in wireless Ad-Hoc network using craft protocol. *Wireless Networks*, *30*(2), 711-735.

[16] Singh, A., Ehtesham, A., Kumar, S., & Khoei, T. T. (2025). A Survey of the Model Context Protocol (MCP): Standardizing Context to Enhance Large Language Models (LLMs).

[17] Hou, X., Zhao, Y., & Wang, H. (2025). The Next Frontier of LLM Applications: Open Ecosystems and Hardware Synergy. arXiv preprint arXiv:2503.04596.

[18] Ghosh, D. (2025). Beyond MCP/A2A: Core LLM limitafions and the future of agenfic AI in the construcfion industry.

[19] PRAVEEN, S. P., ANUSHA, P. V., AKARAPU, R. B., KOCHARLA, S., PENUBAKA, K. K. R., SHARIFF, V., & DEWI, D. A. (2025). AI-POWERED DIAGNOSIS: REVOLUTIONIZING HEALTHCARE WITH NEURAL NETWORKS. *Journal of Theoretical and Applied Information Technology*, *103*(3).

[20] Tirumanadham, N. S. K. M. K., Sekhar, T., & Muthal, S. (2024). An analysis of diverse computational models for predicting student achievement on e-learning platforms using

machine learning. International Journal of Electrical and Computer Engineering (IJECE), 14(6), 7013. https://doi.org/10.11591/ijece.v14i6.pp7013-7021

[21] S, S., Kodete, C. S., Velidi, S., Bhyrapuneni, S., Satukumati, S. B., & Shariff, V. (2024f). Revolutionizing Healthcare: A Comprehensive Framework for Personalized IoT and Cloud Computing-Driven Healthcare Services with Smart Biometric Identity Management. Journal of Intelligent Systems and Internet of Things, 13(1), 31–45. https://doi.org/10.54216/jisiot.130103

[22] Shariff, V., Paritala, C., & Ankala, K. M. (2025b). Optimizing non small cell lung cancer detection with convolutional neural networks and differential augmentation. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-98731-4

[23] K. V. Rajkumar, K. Sri Nithya, C. T. Sai Narasimha, V. Shariff, V. J. Manasa and N. S. Koti Mani Kumar Tirumanadham, "Scalable Web Data Extraction for Xtree Analysis: Algorithms and Performance Evaluation," *2024 Second International Conference on Inventive Computing and Informatics (ICICI)*, Bangalore, India, 2024, pp. 447-455, doi: 10.1109/ICICI62254.2024.00079.

[24] Swapna Donepudi et al., "Security Model For Cloud Services Based On A Quantitative Governance Modelling Approach ", Journal of Theoretical and Applied Information Technology, vol. 101, no. 7, 15th April 2023

[25] Thatha, V. N., Donepudi, S., Safali, M. A., Praveen, S. P., Tung, N. T., & Cuong, N. H. H. (2023). Security and risk analysis in the cloud with software defined networking architecture. International Journal of Electrical and Computer Engineering, 13(5), 5550. https://doi.org/10.11591/ijece.v13i5.pp5550-5559

[26] Praveen, S. P., Thati, B., Anuradha, C., Sindhura, S., Altaee, M., & Jalil, M. A. (2023). A novel approach for enhance fusion based healthcare system in cloud computing. Journal of Intelligent Systems and Internet of Things, 12(1), 88–100. https://doi.org/10.54216/jisiot.090106

[27] Praveen, S. P., Ghasempoor, H., Shahabi, N., & Izanloo, F. (2023b). A hybrid gravitational emulation Local Search-Based algorithm for task scheduling in cloud computing. Mathematical Problems in Engineering, 2023(1). https://doi.org/10.1155/2023/6516482

[28] S. P. Praveen, S. Sindhura, A. Madhuri and D. A. Karras, "A Novel Effective Framework for Medical Images Secure Storage Using Advanced Cipher Text Algorithm in Cloud Computing," *2021 IEEE International Conference on Imaging Systems and Techniques (IST)*, Kaohsiung, Taiwan, 2021, pp. 1-4, doi: 10.1109/IST50367.2021.9651475.

[29] Abuowaida, Suhaila, et al. "Evidence Detection in Cloud Forensics: Classifying Cyber-Attacks in IaaS Environments using machine learning." (2025).

[30] Mohammad, Anber Abraheem Shlash, et al. "Cloud Computing Adoption in the Digital Era: A Bibliometric Analysis and Research Agenda." Artificial Intelligence, Sustainable Technologies, and Business Innovation: Opportunities and Challenges of Digital Transformation (2025): 137-148.

[31] N S Koti Mani Kumar Tirumanadham et al., " Boosting Student Performance Prediction In E-Learning: A Hybrid Feature Selection And Multi-Tier Ensemble Modelling Framework With Federated Learning ", Journal of Theoretical and Applied Information Technology, vol. 103, no. 5, March 2025.

[32] Mohammad, A. A. S., et al. "Data security in digital accounting: A logistic regression analysis of risk factors." International Journal of Innovative Research and Scientific Studies 8.1 (2025): 2699-2709