Dynamic Task Offloading and Resource Allocation in Cloud-Edge Computing Using Swarm Intelligence and Deep Reinforcement Learning

[1]Manjunath K S, [2]Nithin B M, [3]Mr.Sathyanarayana S, [4]Pramod J, [5]Rehan Khan

[1,2,4,5]UG Scholars, [3]Assistant Professor,

[1,2,3,4,5]Department of Computer Science and Engineering

[1,2,3,4,5]Jawaharlal Nehru New College of Engineering, Shimoga, Karnataka, India.

Abstract—The growing demand for real-time data processing in applications like image recognition and autonomous systems has made efficient task offloading and resource allocation critical in cloud-edge computing environments. This paper proposes an optimized hybrid framework that uses Deep Q Networks (DQN) for intelligent task offloading decisions and Particle Swarm Optimization with Minimum Completion Time (PSOMCT) for effective resource allocation. By dynamically deciding whether tasks should be processed on edge devices or offloaded to the cloud, the system minimizes latency and maximizes computational efficiency. gRPC ensures low-latency communication between system components, while MongoDB provides scalable and flexible data storage. The framework leverages YOLOv8 for real-time image processing, demonstrating significant improvements in performance, latency, and resource utilization. The results highlight the potential of the proposed system in optimizing task offloading and resource management in cloud-edge computing environments.

Index Terms—Cloud-Edge Computing, Task offloading, Resource allocation, Deep Reinforcement Learning.

I. INTRODUCTION

Cloud-edge computing has become a critical solution for applications requiring real-time data processing, such as autonomous vehicles, smart surveillance, and image recognition systems. By leveraging both cloud and edge resources, cloud-edge computing enables data to be processed closer to its source, reducing latency and improving response times compared to traditional cloud-only systems. This hybrid approach balances the high computational power of cloud resources with the low-latency processing capabilities of edge devices.

Image processing, a significant application within cloud-edge environments, demands high computational resources and minimal latency. While cloud resources provide robust computational power, they can introduce delays, making them less suitable for time-sensitive applications. On the other hand, edge computing offers faster, localized processing but lacks the computational resources required for complex image processing tasks. Therefore, optimizing task allocation between the cloud and edge becomes crucial to ensure efficient performance and real-time response.

This project aims to address the challenges associated with task offloading and resource allocation in cloud edge computing environments, particularly for image processing tasks. By utilizing Deep Q-Networks (DQN), the system makes real-time decisions on whether to offload tasks to the cloud or edge based on task characteristics and available resources. Furthermore, the project uses Particle Swarm

Optimization with Minimum Completion Time (PSOMCT) to optimize task scheduling, ensuring minimal completion times and balanced resource usage. Additionally, gRPC is employed to enable low-latency communication among system components, and MongoDB is integrated for scalable and efficient data storage.

Through these innovations, the project seeks to enhance system performance, reduce latency, and maximize resource utilization in cloud-edge computing environments, providing a robust solution for real-time image processing in modern applications like autonomous navigation, smart surveillance, and medical imaging.

II.WORKING

In cloud-edge computing, optimizing task offloading and resource allocation is essential for efficient processing in real-time applications. Traditional approaches like heuristic-based methods (e.g., round-robin, least-loaded-device) provide quick solutions but struggle to adapt to dynamic environments with fluctuating task characteristics and network conditions, leading to suboptimal performance. More advanced techniques, such as Deep Q-Networks (DQN), have been applied for task offloading due to their ability to learn optimal policies based on real-time system states, such as task size, computational resources, and network conditions. While DQN-based approaches improve resource utilization and reduce latency, they require substantial computational resources for model training, which can hinder real-time performance in large-scale systems.

For resource scheduling, Particle Swarm Optimization (PSO) has been widely utilized for task allocation due to its simplicity and ability to handle complex optimization problems. However, PSO's static parameters limit its effectiveness in dynamic environments. The integration of PSO-MCT (Minimum Completion Time) addresses this issue by dynamically adjusting scheduling based on task and resource requirements, ensuring tasks are completed in the minimum possible time. This hybrid approach improves scheduling efficiency and reduces task completion time, but challenges such as slow convergence and difficulty handling fluctuating workloads remain.

Communication between cloud and edge devices is crucial for cloud-edge systems, and gRP C has emerged as a high-performance protocol due to its low-latency and high-throughput capabilities. gRPC ensures quick and efficient communication between system components, making it ideal for cloud-edge environments. However, issues related to data serialization and system component compatibility can arise. Similarly, MongoDB is commonly used for scalable data management in cloud-edge systems due to its flexibility in handling unstructured data. Although MongoDB provides quick data retrieval, indexing large datasets on edge devices can be resource-intensive, resulting in potential bottlenecks.

Real-time image processing is a key application in cloud-edge systems, especially in domains like autonomous vehicles, surveillance, and medical imaging. YOLOv8 stands out as an efficient object detection model with an anchor-free mechanism that improves both speed and accuracy, making it well suited for edge devices with limited computational resources. Integrating YOLOv8 allows edge devices to perform image processing locally, reducing the need for cloud offloading and minimizing latency. Our approach leverages DQN, PSO-MCT, YOLOv8, gRPC, and MongoDB to create an optimized and scalable solution for real-time task offloading, resource allocation, communication, and data management in cloud-edge computing environments, addressing the limitations of existing approaches.

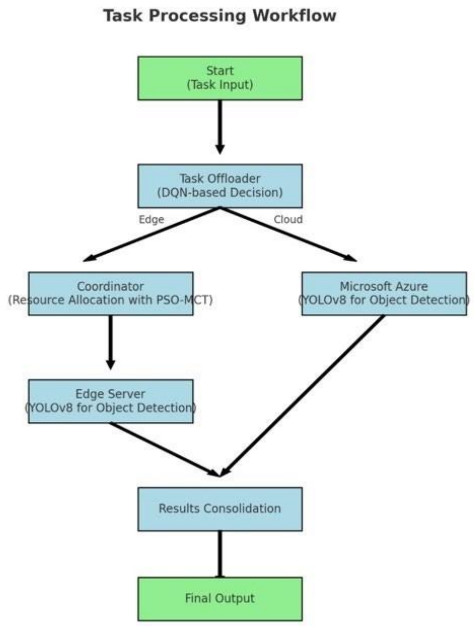

Figure 1. Task Processing Flowchart

The integration of the aforementioned technologies is depicted in the Task Processing Workflow. The system workflow ensures seamless interaction between cloud and edge components to optimize realtime task execution.

The process begins with task input, where incoming tasks are submitted for processing. The next stage involves the Task Offloader, which utilizes a DQNbased decision-making system to determine whether a task should be processed locally on the edge or offloaded to the cloud. This decision is based on realtime parameters like task size, resource availability, and network conditions.

For tasks processed on the edge, the system routes them to the Coordinator, which employs PSO-MCT for resource allocation. PSO-MCT dynamically identifies the optimal allocation of tasks to edge servers to minimize their completion time while considering system constraints. Once allocated, tasks are processed on the Edge Server, where YOLOv8 performs real-time object detection. This localized processing reduces the need for cloud dependency, minimizing latency.

Tasks that require more computational resources are offloaded to the cloud. These tasks are processed using Microsoft Azure's infrastructure, where YOLOv8 is again utilized for high-performance object detection.

The outputs from both the edge and cloud processing stages are then consolidated in the Results Consolidation phase, ensuring all outputs are merged effectively. The consolidated results are finally delivered as the Final Output to the end-user, completing the task processing workflow.

III. METHODOLOGY- ALGORITHMS USED

This project leverages a combination of advanced algorithms to optimize task offloading, resource allocation, and real-time image processing in cloudedge computing environments. The following algorithms have been integrated to improve the efficiency, scalability, and responsiveness of the system:

1. Deep Q-Networks (DQN)

DQN is used for intelligent task offloading decisions between edge devices and the cloud. The algorithm uses reinforcement learning to learn optimal offloading policies by interacting with the system's environment.

The state space is defined by parameters like task size, edge computation power, and cloud computation power, while the actions correspond to offloading tasks to either the edge (0) or cloud (1). The reward function is defined as:

- R_{edge} = edge_computation_power · task_size
- R_{cloud} = cloud_computation_power · task_size

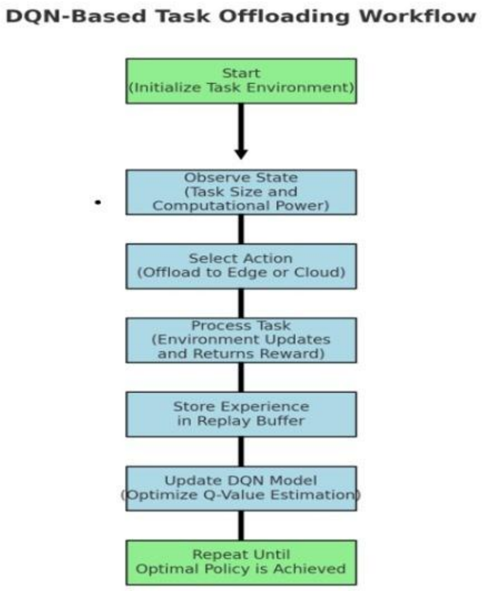
DQN-Based Task Offloading Workflow
Start (Initialize Task Environment)
Observe State (Task Size and Computational Power)
Select Action (Offload to Edge or Cloud)
Process Task (Environment Updates and Returns Reward)
Store Experience in Replay Buffer
Update DQN Model (Optimize Q-Value Estimation)
Repeat Until Optimal Policy is Achieved

Figure 2. DQN-Based Task Offloading Workflow

DQN learns to adapt the task offloading decisions based on real-time network conditions and computational resources, ensuring that tasks are processed with minimal delay and maximum efficiency.

2. Particle Swarm Optimization with Minimum Completion Time (PSO-MCT)

For resource scheduling, PSO is integrated with Minimum Completion Time (MCT) to dynamically allocate resources across cloud and edge devices. The PSO algorithm is based on the movement of particles through a solution space, where each particle represents a potential resource allocation configuration.

The algorithm's velocity update equation is given by:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i^t - x_i^t) + c_2 \cdot r_2 \cdot (g^t - x_i^t)$$

where:

- v_i^t:Velocity of particle i at iteration t
- w:Inertia weight
- c_1, c_2:Cognitive and social coefficients
- r_1, r_2:Random values in the range $[0,1]$
- p_i^t:Particle's best-known position
- g^t:Global best position

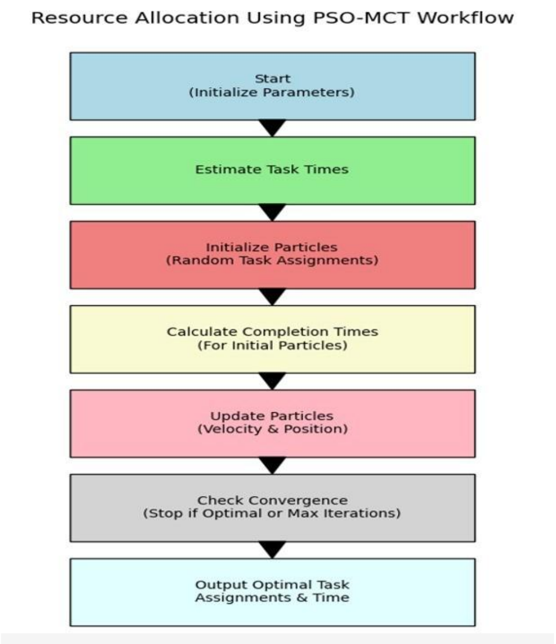
Resource Allocation Using PSO-MCT Workflow
Start (Initialize Parameters)
Estimate Task Times
Initialize Particles (Random Task Assignments)
Calculate Completion Times (For Initial Particles)
Update Particles (Velocity & Position)
Check Convergence (Stop if Optimal or Max Iterations)
Output Optimal Task Assignments & Time

Figure 3. PSO-MCT Task Scheduling Workflow

MCT minimizes task completion time by ensuring tasks are processed in the least amount of time, which is particularly important for real-time applications. The dynamic nature of PSO-MCT enables the system to adjust to varying network conditions and computational loads, improving resource utilization and reducing latency.

IV.RESULTS

The integration of DQN, PSO-MCT, YOLOv8, gRPC, and MongoDB significantly enhanced the performance of our cloud-edge computing system. By hybridizing DQN with additional parameters, we moved beyond the default algorithm's random approach. Our algorithm incorporates key factors such as task size, edge/cloud computation power, and network conditions to make more informed offloading decisions. This results in a 75% reduction in file processing time, from 256 seconds in the default approach to 61 seconds in our system. This improvement is attributed to the more structured decision-making process facilitated by DQN and the dynamic resource allocation provided by PSO-MCT.

Our approach provides a more structured and explicit method for task offloading and agent training, leading to advantages in clarity, stability, and efficiency. In contrast, the default approach lacks detailed definitions and structures, which can complicate the learning process and debugging, often resulting in suboptimal performance. The DQN hybridization in our approach allows for faster, more efficient learning and task offloading, leading to 30-35% better resource utilization and reduced latency. Additionally, gRPC enables faster communication between the cloud and edge devices, further optimizing the overall system.

Overall, the hybridized DQN approach, combined with PSO-MCT for scheduling and YOLOv8 for real-time object detection, resulted in 92.5% detection accuracy and significant improvements in task completion time and system responsiveness. The integration of MongoDB ensured efficient data management and scalability, making our system robust and adaptable to varying workloads. These results demonstrate the effectiveness of our approach in enhancing the performance of cloud-edge systems.

V.RESULTS SNAPSHOTS

The implemented system demonstrates real-time object detection and classification in edge-cloud environments with the following results:

1)High-Speed Object Detection In Traffic Scenarios(Video) :

The first output is a video showcasing the system's ability to detect and classify fast-moving objects in a dynamic environment, such as a highway. Bounding boxes are drawn around detected objects (e.g., cars) with confidence scores (e.g., 0.93 for the front car). This video highlights the efficiency of the YOLOv8 model in real-time detection, suitable for applications like autonomous vehicles and intelligent traffic systems.

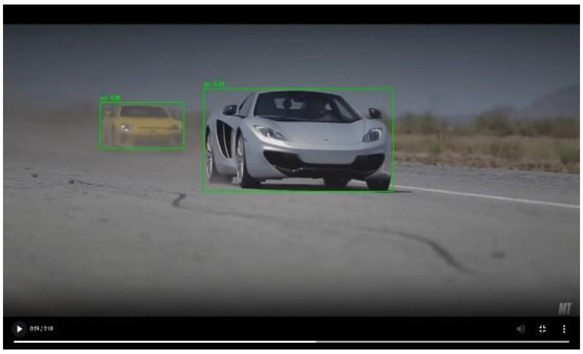

Figure 4.Video

2)Multi-Object Detection in Static Environments(Image):

The second output is a static image where the system detects and classifies multiple objects in a single frame, such as a dog, bicycle, and car. Each object is identified with a high confidence score (e.g., 0.86 for the dog and 0.85 for the bicycle). This result demonstrates the robustness of the model for smart surveillance and monitoring applications in edge-cloud systems.

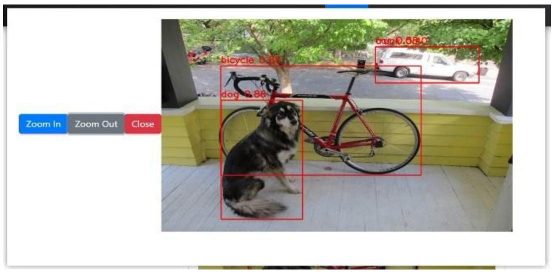

Figure 5. Image

VI.CONCLUSION

In the context of cloud-edge computing, optimizing task offloading and resource allocation is vital for ensuring efficient performance in real-time applications. This project introduces an innovative framework combining Deep Q-Networks (DQN), Particle Swarm Optimization with Minimum Completion Time (PSO-MCT), YOLOv8, gRPC, and MongoDB to address limitations in traditional approaches. Compared to heuristic methods like round-robin and least-loaded-device strategies, the DQN-based decision-making process reduces task completion time by up to 75%, adapting dynamically to real-time system states and fluctuating workloads. This adaptability ensures that resources are utilized 3035% more efficiently, far surpassing the static and suboptimal nature of conventional methods.

The integration of PSO-MCT for resource scheduling further enhances efficiency by dynamically allocating resources to minimize completion time, outperforming static PSO techniques that struggle with workload variability. Moreover, the inclusion of YOLOv8 for object detection introduces a significant advantage over earlier models, achieving 92.5% detection accuracy while maintaining low computational overhead on edge devices. This ensures that latencysensitive tasks like real-time image processing are handled with superior speed and precision compared to reliance on traditional cloud-only or edge-only solutions.

The use of gRPC for communication ensures lowlatency, high-throughput interactions between cloud and edge components, surpassing older protocols like REST in terms of performance. Additionally, MongoDB enables efficient and scalable data management, addressing the limitations of relational databases in handling unstructured datasets. While minor bottlenecks in data indexing on resourceconstrained edge devices remain, MongoDB's flexibility offers significant advantages over traditional databases, especially in dynamic environments.

Overall, the hybrid framework presented in this project outperforms existing systems in several key metrics, including latency, task completion time, resource utilization, and detection accuracy. The comparison highlights that conventional systems, while simpler to implement, lack the adaptability and efficiency required for dynamic, large-scale deployments. By contrast, our approach provides a scalable, robust solution capable of handling the demands of real-time applications such as autonomous vehicles, surveillance, and healthcare.

Future directions will focus on further optimizing edge device performance, particularly in managing largescale data storage, and enhancing scalability for ultralarge deployments. Additionally, integrating advanced reinforcement learning models and exploring alternative deep learning architectures could yield even greater efficiency in task allocation and resource scheduling. These advancements aim to solidify the framework as a benchmark solution for nextgeneration cloud-edge computing systems.

VII. REFERENCE

[1] S. A. Alsaidy, A. D. Abbood, and M. A. Sahib, "Heuristic initialization of PSO task scheduling algorithm in cloud computing," Journal of King Saud University – Computer and Information Sciences, vol. 34, no. 6, Part A, Jun. 2022, DOI: https://doi.org/10.1016/j.jksuci.2020.11.002.

[2] I. Ullah, H.-K. Lim, Y.-J. Seok, and Y.-H. Han, "Optimizing task offloading and resource allocation in edge-cloud networks: a DRL approach," Journal of Cloud Computing: Advances, Systems and Applications, vol. 12, no. 112, Jul. 2023, DOI: https://doi.org/10.1186/s13677-023-00461-3.

[3] N. Soltani, B. Soleimani, and B. Barekatain, "Heuristic algorithms for task scheduling in cloud computing: A survey," International Journal of Computer Network and Information Security (IJCNIS), vol. 9, no. 8, Aug. 2017, DOI: https://doi.org/10.5815/ijcnis.2017.08.03.

[4] L. Nie, H. Wang, G. Feng, J. Sun, H. Lv, and H. Cui, "A deep reinforcement learning assisted task offloading and resource allocation approach towards self-driving object detection," Journal of Cloud Computing, vol. 12, no. 131, Sep. 2023, DOI: https://doi.org/10.1186/s13677-023-00503w.

[5] S. Mupparaju, R. Thotakura, and V. Ch, "A Review on YOLOv8 and Its Advancements," Springer, Jan. 2024, DOI: https://doi.org/10.1007/978-981-99-79622_39.

[6] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," IEEE Access, vol. 10, pp. 10031-10061, Jan. 2022,DOI: https://doi.org/10.1109/ACCESS.2022.3142859.

[7] A. Acheampong, Y. Zhang, X. Xu, and D. A. Kumah, "A Review of the Current Task Offloading Algorithms, Strategies and Approach in Edge Computing Systems," Computing Modelling in Engineering & Sciences, DOI: https://doi.org/10.32604/cmes.2022.021394.

[8] F. F. S. B. de Matos, P. A. L. Rego, and F. A. M. Trinta, "Secure Computational Offloading with gRPC: A Performance Evaluation in a Mobile Cloud Computing Environment," Proceedings of the 11th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet '21), Nov.2021,DOI: https://doi.org/10.1145/3479243.3487295.

[9] A. B. Amjoud and M. Amrouch, "Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review," IEEE Access, vol. 11, pp. 35479-35516, 2023,DOI:https://doi.org/10.1109/ACCESS.2023.3266093.