HDGC-hybrid task offloading framework using deep reinforcement learning and genetic algorithms for 6G edge cloud

Kaniezhil Radhakrishnan[1], Mong-Fong Horng[2], Siva Shankar Subramanian[3], Chun-Chih Lo[2]
[1]Department of Computer Science, Navarasam Arts and Science College for Women, Tamil Nadu, India
[2]Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan
[3]Department of Computer Science and Engineering, KG Reddy College of Engineering and Technology, Hyderabad, India

Article Info

Article history:

Received Oct 17, 2024
Revised Jan 31, 2026
Accepted Feb 17, 2026

Keywords:

6G networks
Deep reinforcement learning
Edge computing
Genetic algorithms
Task offloading

ABSTRACT

The rapid evolution of 6G networks has brought new challenges in the domain of task offloading (TO), particularly within edge computing environments that are heavily reliant on the internet of things (IoT). Traditional TO methods that based on rule-based heuristics or shallow learning techniques fail to adapt efficiently to the dynamic, unpredictable network conditions, resource heterogeneity, and varying task demands. The proliferation of edge computing, the IoT, and 6G networks has introduced new challenges in TO due to dynamic network conditions, resource heterogeneity, and unpredictable task demands. To address these challenges, this work proposes an innovative TO method that integrates deep reinforcement learning (DRL) with heuristic search methods. The combination of DRL and heuristic algorithms enhances adaptability, convergence speed, and decision-making efficiency, making it well-suited for real-time TO in complex and unpredictable environments This paper proposes a novel hybrid TO framework that integrates DRL with genetic algorithms (GA) to address these challenges. The proposed hybrid optimization technique offer promising solutions by leveraging the strengths of individual approaches to balance competing objectives, such as energy consumption, task completion time, and resource utilization. This method explores optimization strategies to enhance TO efficiency in decentralized environments mainly focusing on optimizing energy use while ensuring performance metrics like latency, throughput, and task deadlines are met.

Corresponding Author:

Kaniezhil Radhakrishnan
Department of Computer Science, Navarasam Arts and Science College for Women
Arachalur, Tamil Nadu, India
Email: kaniezhil@yahoo.co.in

1. INTRODUCTION

The evolution of mobile and wireless communication networks has witnessed a significant leap with the deployment of 5G, enabling ultra-reliable low-latency communications (URLLC), enhanced mobile BroadB and (eMBB), and massive machine-type communications (mMTC). In the realm of edge computing and the internet of things (IoT), 5G has facilitated intelligent task offloading (TO) by bringing computation closer to the data source, thereby diminishing response time and network congestion. Existing methodologies for TO in 5G-enabled environments predominantly rely on rule-based heuristics, static optimization, or shallow learning models that consider parameters such as latency, energy consumption, bandwidth, and computation capacity [1]-[5]. However, these techniques often fail to scale effectively under highly dynamic conditions, particularly when network topologies change rapidly, resource availability fluctuates, or application workloads are non-stationary. Furthermore, 5G-based frameworks struggle with cross-layer optimization due to the limited contextual awareness of device states and task dependencies, leading to sub-optimal allocation of resources and increased system overhead. Traditional reinforcement learning (RL) approaches employed within 5G systems, while adaptive often face convergence issues, exploration-exploitation imbalance, and lack of generalization in non-stationary environments [6]-[9].

As we advance into the 6G era, communication paradigms undergo a transformative shift. 6G networks are characterized by terahertz communication, intelligent surfaces, ultra-massive MIMO, and integration with artificial intelligence (AI-native) infrastructure. These innovations offer unprecedented support for real-time, distributed, and context-aware computation offloading [10]-[14]. The inherent intelligence embedded in 6G architecture enables seamless orchestration among Edge, Fog, and Cloud layers. Moreover, 6G facilitates joint optimization of computation, communication, and control by leveraging high-bandwidth low-latency links, pervasive sensing, and deep semantic understanding of tasks and services. In this landscape, the need for robust, adaptive, and intelligent TO mechanisms becomes paramount to exploit the full potential of 6G-enabled systems [15]-[19].

To address these challenges, this work proposes a novel hybrid TO framework that integrates deep reinforcement learning (DRL) with heuristic genetic algorithms (GA) in a 6G-enabled edge-cloud ecosystem. DRL architectures, particularly actor-critic (AC), and deep Q-networks, are employed to learn dynamic offloading policies that adapt to real-time network states [20]-[24]. To overcome DRL limitations such as slow convergence and policy suboptimality, GA is embedded to refine the exploration process and optimize decision quality [25]-[27]. The proposed model incorporates transfer learning, Transformer-based attention mechanisms for long-term dependency modeling, and graph neural networks (GNNs) to capture the spatial-temporal task dependencies. This hybrid framework significantly enhances adaptability, computational efficiency, and resource utilization under complex and unpredictable network environments. The paper is organized as follows; sections 2 discuss the related work and deines TO strategy. In section 3, the problem of the existing system was discussed and point out how to eradicate the issues in the existing system and thereby improve the energy efficiency. Section 4 discuss the proposed framework. Section 5 presents results and discussion and finally, conclusions are presented in section 6.

2. LITERATURE REVIEW

A comprehensive comparative analysis of energy consumption in mobile edge computing (MEC) environments was presented in [1], where several quality-of-service (QoS) parameters were evaluated in terms of usage efficiency and percentage-based performance. Both full and partial TO strategies were examined using different metaheuristic optimization algorithms, highlighting key limitations of MEC systems such as scalability constraints, computational overhead, and energy inefficiency under dynamic workloads.

Li *et al.* [2] evaluated and validated the effectiveness of a TO algorithm for multiserver edge computing. The study jointly optimized diverse system parameters and presented a comparative analysis against five existing algorithms, demonstrating improved efficiency, load balancing, and performance. An optimal decision-making model for TO and resource allocation was proposed in [3], aiming to minimize execution delay and energy consumption while maximizing computing rates under strict energy constraints. The model enables system maintenance at each time frame using DRL. Furthermore, edge-to-cloud offloading was shown to improve task completion rates by efficiently managing data transfer between edge servers and cloud infrastructures. In general, efficient TO and energy-aware mechanisms significantly enhance MEC server utilization through intelligent load balancing and task scheduling [4]. However, limited battery capacity remains a major challenge when optimizing TO decisions under unreliable edge performance conditions [5].

A computation offloading model over a collaborative cloud–edge network was introduced in [6], considering complete offloading requirements and server resource utilization. The authors formulated the offloading strategy as a convex optimization problem, guaranteeing a unique and optimal solution. In a broader perspective, edge–cloud computing environments effectively bridge the gap between centralized cloud infrastructures and end users by enabling near-data computing services with minimal latency. TO is therefore regarded as one of the most promising approaches to enhance service quality, energy efficiency, and resource utilization in edge–cloud systems [7]. A cooperative TO strategy based on multi-agent deep reinforcement learning (MADRL) was proposed in [8], where each edge node operates as an intelligent agent. Centralized training with decentralized execution mitigates non-stationarity issues and improves TO efficiency by optimizing a shared reward function. The proposed scheme significantly reduces energy consumption while satisfying computation and communication constraints.

DRL techniques such as deep Q-networks (DQN) were shown to provide adaptive and efficient decision-making in dynamic and resource-constrained edge environments [9]. Additionally, hybrid models integrating long short-term memory networks with differential evolution (LSTM-DE) and fuzzy Q-learning were employed to predict future IoT workloads and enable proactive resource auto-scaling and offloading decisions for mobile applications [10]. Near-real-time and highly accurate predictions for optimal offloading decisions and computing resource allocation were achieved using a multi-task learning-based feed-forward neural network (MTFFN) model in [11]. Simulation results demonstrated superior performance compared with benchmark offloading approaches. TO strategies may be classified as binary or partial. Binary offloading executes tasks either locally or fully offloads them to cloudlets, whereas partial offloading distributes tasks between local devices and remote servers. Moreover, offloading decisions can be static or dynamic; static offloading assigns tasks to predefined edge servers, while dynamic offloading adapts decisions based on real-time system conditions [12].

3. TASK OFFLOADING

In fields including AI, machine learning, and big data analysis, TO is crucial for enabling communication between local devices and cloud infrastructure. The need for devices that do not have substantial computer power on their own serves as the foundation for its importance. These methods transfer calculations from potentially resource-constrained devices, including mobile phones and IoT devices in cloud computing, to cloud or edge servers. The cloud services are enabled in a such a way that a task to be done flexibly in a given time without overloading local applications. These approaches are to enhance the performance, minimize the delay and regulate power usage. This means that organizations can avoid expending large sums in the purchase of hardware and other support structures by hiring cloud resources on as-needed basis. TO is a technique used to improve the distribution of computing activities among several devices, such as cloud servers, edge devices, or mobile phones, especially in the context of energy conservation. The goal of TO is to improve performance and reduce overall energy consumption, especially in systems with limited resources, such battery-operated devices. The objective is to shift processing-heavy tasks to the edge, near the data source, in order to minimize latency, conserve bandwidth, and optimize resource usage, all while maintaining energy efficiency.

The approaches are to enhance the performance, minimize the delay and regulate power usage. This means that organizations can avoid expending large sums in the purchase of hardware and other support structures by hiring cloud resources on as-needed basis. Key concepts of TO: i) edge nodes, ii) energy efficiency, iii) latency reduction, iv) bandwidth optimization, and v) scalability.

3.1. Problem definition

In this ubiquitous age, real-time data processing and low latency responses have become essential. Traditional cloud computing systems that concentrate data processing in distinct data centers often fall short of these requirements due to intrinsic latency and bandwidth constraints. While offloading preferences specific to a user, such as desired latency or willingness to expend energy, are taken into consideration in offloading decisions, existing offloading strategies are usually modelled so that they can be applied to all users. The proposed arguments explained to open challenges:

a. Outsourcing tasks on to a cloud primarily depends on network quality and high latency hampers real-time or time-constrained tasks.
b. The offloading of sensitive tasks increases questions over data privacy and security.
c. Resource management is a challenge, especially when operating over multiple cloud domains.

Thus, the rectification of the mentioned challenges in offloading forms the goal of this study to enhance the offloading performance and reduce on energy consumption and cost. The edge cloud computing paradigm, which moves computation closer to the data source to increase data processing efficiency, is examined in this paper. By using edge devices such as sensors, gateways, and mobile devices that are located near or at the data generating sites, edge computing significantly reduces latency and bandwidth consumption. By reducing the load on centralized data centers and providing real-time processing capabilities, this approach enhances the scalability and reliability of systems.

4. THE PROPOSED FRAMEWORK

Traditional TO methods rely heavily on rule-based heuristics or shallow learning models. These approaches lack the flexibility to handle the unpredictable and dynamic nature of 6G networks. Traditional methods often result in suboptimal resource allocation and increased latency. They also suffer from poor scalability in complex network environments. RL techniques, while adaptive, face challenges such as slow convergence. They are also prone to getting trapped in local optima, thus limiting their effectiveness. There is a pressing need for hybrid frameworks to overcome the limitations of existing methods. These frameworks aim to combine adaptive learning capabilities with heuristic optimization methods. There comes a solution for the existing challenges are hybrid frameworks leverage DRL and GA for better decision-making. This combination enables a more efficient exploration-exploitation balance in complex edge-cloud ecosystems.

The hybrid deep reinforcement learning with genetic computation (HDGC) framework combines DRL and GA. It is designed to optimize TO in 6G-enabled edge-cloud networks. GA enhances the exploration process through population-based search. This accelerates convergence and helps avoid local optima in TO. The framework incorporates transfer learning to adapt to new environments quickly. This ensures robust, scalable, and context-aware task distribution across edge and cloud layers. The synergy of DRL and GA in the HDGC framework optimizes TO. It achieves efficient and adaptive performance in complex 6G network scenarios.

4.1. Network model

The proposed network model is designed for a 6G-enabled hierarchical edge-cloud computing architecture, comprising three primary layers: IoT device layer, edge computing layer, and cloud data center layer. At the lowest tier, a diverse set of IoT devices generates computational tasks characterized by varying sizes, energy budgets, latency requirements, and data dependencies. Figure 1 display the network model of the work.

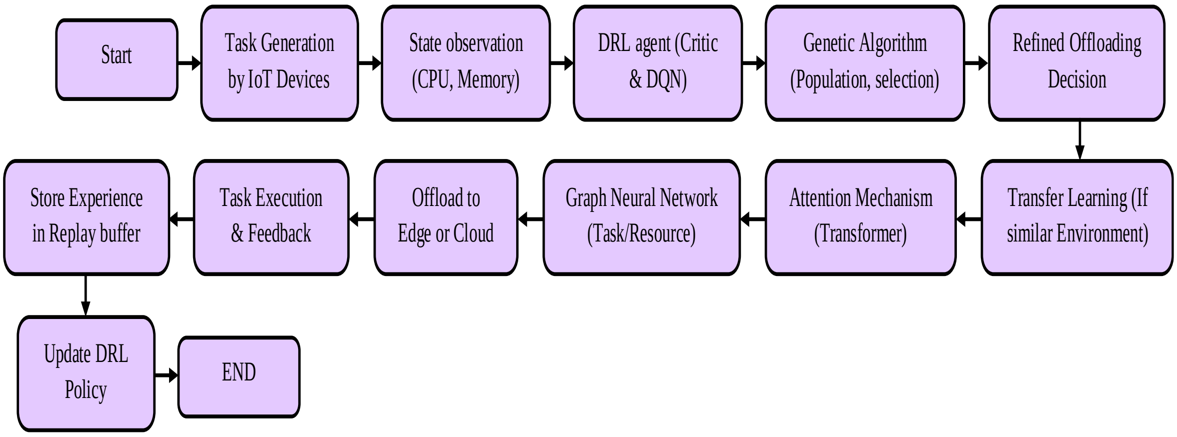
Start
Task Generation by IoT Devices
State observation (CPU, Memory)
DRL agent (Critic & DQN)
Genetic Algorithm (Population, selection)
Refined Offloading Decision
Transfer Learning (If similar Environment)
Attention Mechanism (Transformer)
Graph Neural Network (Task/Resource)
Offload to Edge or Cloud
Task Execution & Feedback
Store Experience in Replay buffer
Update DRL Policy
END

Figure 1. HDGC architecture

These devices are connected to nearby edge servers through ultra-reliable low-latency 6G links, which support dynamic spectrum access and intelligent reflecting surfaces to ensure stable and high-throughput connections. The edge layer consists of heterogeneous edge servers with differing computational capacities and energy profiles, forming a decentralized and cooperative processing network.

Each edge node periodically senses its environment and collaborates with adjacent nodes to share resource availability and workload information via 6G's high-speed device-to-device (D2D) communication. A central DRL agent, embedded within the edge layer, observes global state parameters—such as bandwidth utilization, CPU and memory status, transmission delays, and task graphs—and generates optimal TO decisions using an AC architecture. To enhance the decision process, the DRL policy is augmented with GA that introduce population-based refinements, thus accelerating convergence and preventing suboptimal local decisions. In this work, we propose a novel hybrid framework that combines DRL with heuristic optimization, particularly GA for dynamic and intelligent TO from edge servers to the cloud. This section delineates the structural components of the proposed method, elucidates their interactions, and presents detailed mathematical formulations used in the development and integration of the framework.

4.2. Deep reinforcement learning-based task offloading

At the core of the proposed method is a DRL-based decision engine designed to optimize TO decisions in real-time as shown in Figure 2. Let the system state at time t be represented as $s_t \in \mathbb{R}^n$, which captures features such as CPU utilization, memory usage, network latency, and energy levels of edge and cloud resources. The action $a_t \in A$ corresponds to the decision of whether to offload a given task T_i to a local edge node E_j, another edge node, or to the cloud server C.

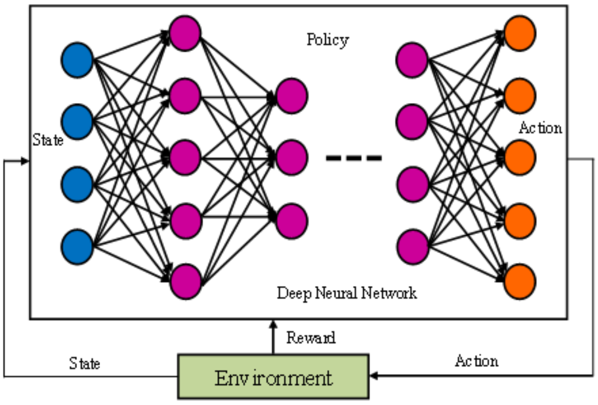

Figure 2. DRL method

The goal of the reward function R(st, at) is to minimize delay and energy usage while optimizing job throughput. The DRL agent seeks to maximize the predicted cumulative discounted reward by learning an optimum strategy π(at| st):

$$E_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]=J(\pi) \quad (1)$$

where the discount factor is denoted by $\gamma \epsilon [0, 1]$.

We adopt an AC architecture where the actor network π_θ proposes actions, and the critic network $V_\phi(s_t)$ estimates the value function:

$$V_\phi(s_t) = E_\pi\left[\sum_{t=0}^{\infty} \gamma^k \ R(s_{t+k}, a_{t+k})\right] \quad (2)$$

Gradients for the actor are computed using the advantage function:

$$\nabla_\theta J(\pi) = E_t[\nabla_\theta \log\pi_\theta a_t| \ s_t)A_t]$$

$$A_t = R(s_t, a_t) + V_\phi(s_{t+1}) - \nabla_\theta(s_t) \quad (3)$$

4.3. Task offloading between edge and cloud

In the multi-layer computing environment, each task T_i is characterized by its size D_i, computation requirement C_i, and deadline L_i. The system must decide the best location for processing: edge node E_j, peer edge node E_k, or cloud server C. The decision vector a = [a_1, a_2, ..., a_N] maps each task to a target compute node.

The offloading decision affects task execution delay τ_i, which includes transmission time τ_{trans} and execution time τ_{exec}:

$$\tau_i = \tau_{trans}(T_i, a_i) + \tau_{exec}(T_i, a_i) \quad (4)$$

The total system latency L_{total} is (5).

$$L_{total} = \sum_{i=1}^{N} \tau_i . I(\tau_i \leq L_i) + \infty . I(\tau_i > L_i) \quad (5)$$

Where I is the indicator function enforcing deadline constraints. The reward function incorporates a penalty for missed deadlines and energy overuse.

DRL with deep Q-Learning for state-action estimation in parallel, a DQN is trained to estimate Q-values of state-action pairs:

$$Q(s_t, a_t) = E_\pi[R(s_t, a_t) + \max_{a'} Q(s_{t+!}, a') \quad (6)$$

To stabilize learning, experience replay and target networks are employed. The loss function minimized at each iteration is:

$$L_\theta = E_{(s,a,r,s')}[(r + \max_{a'} Q_\theta(s',a') - Q_\theta(s,a)^2] \tag{7}$$

Here, Q_θ is a periodically updated target network.

4.4. Incorporation of genetic algorithms

To improve exploration-exploitation and avoid local optima, a GA is employed alongside DRL. Each candidate solution a is encoded as a chromosome representing offloading decisions. The fitness function F(a) is defined based on reward R:

$$F(a) = \sum_{i=1}^{N}\left(\omega_1.\frac{1}{\tau_i} + \omega_2.\left(1 - \frac{E_i}{E_{max}}\right)\right).I\,(\tau_i \le L_i) \tag{8}$$

Genetic operations include selection, crossover, and mutation as shown in Figure 3.

- Selection: high-performing chromosomes are chosen for reproduction, propagating beneficial traits.
- Crossover: genetic material is exchanged between selected chromosomes to create novel solutions.
- Mutation: random alterations are introduced to maintain diversity and prevent premature convergence.

The GA improves the offloading policy with each DRL iteration. It generates a population of alternative offloading decisions, selecting the fittest solutions to inform subsequent DRL training phases. This iterative process ensures continuous improvement and adaptation of offloading strategies, leading to more robust and globally optimal solutions.

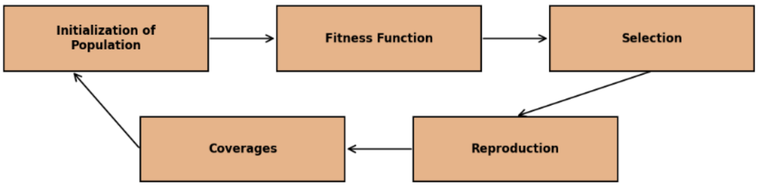

Figure 3. Genetic process

4.5. Transfer learning for adaptive offloading

Transfer learning enables the use of pre-trained DRL policies to accelerate learning in new but related environments. Suppose π_θ^{source} is the source policy trained under known network conditions. For a new task environment E_target, parameters θ are initialized as (9):

$$\theta_{target}\theta_{source} + \Delta\theta \tag{9}$$

where Δθ is adjusted via online learning to fine-tune the policy in E_target.

4.6. Attention-based reinforcement learning

To capture long-term task dependencies, attention-based RL mechanisms using Transformer models are integrated. The state sequence $s_t = [s_{t-k,\dots\dots} s_t]$ is processed through a self-attention mechanism:

$$\text{Attention(Q, K, V)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{10}$$

Here, Q, K, V are queries, keys, and values derived from s_t, and d_k is the dimension of the key vectors. The attention output serves as a context-aware representation for action selection.

4.7. Graph neural networks for dependency modeling

To represent and process the dependencies among tasks and nodes, GNNs are utilized. Let the dynamic system be modeled as a directed graph G=(V, E), where nodes represent tasks or devices, and edges represent communication or dependency links.

The feature of each node $v_i \epsilon V$ is updated as (11):

$$h_i^{(l+1)} = \sigma(\sum_{j \in N(i)} W^{(l)} h_j^{(l)} + b^{(l)}) \tag{11}$$

where N(i) is the neighborhood of node i, $W^{(l)}$ is the weight matrix at layer l, and σ is an RELU activation function. These embeddings are concatenated into the state vector used by the DRL agent.

4.8. Hybrid learning cycle

The training process involves the following steps:

a. The DRL agent observes state s_t and selects action a_t.
b. GA generates a candidate population of offloading strategies around a_t.
c. Fitness evaluation and selection are performed.
d. DRL policy is updated based on the refined action.
e. Experience is stored and replayed for stability.
f. Transfer learning initializes or updates the model in new environments.
g. Transformer and GNN modules enhance temporal and structural understanding.

This hybrid approach enables rapid adaptation to evolving network conditions, reduces convergence time, and significantly enhances the overall efficiency of TO in edge-cloud integrated networks

5. RESULTS AND DISCUSSION

TO in energy efficiency, particularly in edge and cloud computing, involves transferring computational tasks to remote servers or edge nodes to optimize power consumption and performance. Several performance metrics can be used to evaluate the efficiency of TO strategies. These metrics typically focus on energy consumption, execution time, resource utilization, and overall system efficiency. Below are some key performance metrics:

- Energy efficiency metrics (energy consumption),
- Execution performance metrics (throughput),
- Network performance metrics (delay),
- Resource utilization metrics,
- QoS metrics.

The metrics are used for ensuring the optimal allocation of resources across edge nodes, network connectivity and network performance impacts energy efficiency, and helps to evaluate the efficiency of task execution after offloading. The HDGC framework is evaluated using critical metrics for 6G edge-cloud networks.

5.1. Packet delivery ratio

The packet delivery ratio (PDR) is considered as the main performance parameter which provides the assess the efficacy of the suggested hybrid DRL and HDGC offloading paradigm. PDR depicted to be a crucial measure of network reliability and efficiency based on the successful ratio of the delivered data packets towards all packets broadcast. In the context of edge-cloud offloading over 6G-enabled networks, high PDR directly correlates with the robustness of routing and offloading decisions under dynamic and heterogeneous conditions as shown in Figure 4.

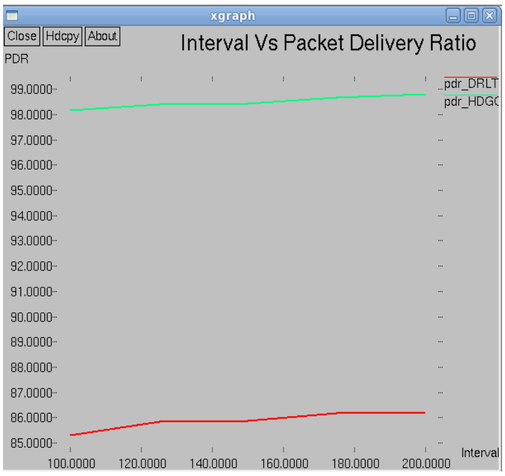

Figure 4. Interval vs PDR

5.2. Throughput

Throughput, defined as the total volume of successfully transmitted data over the network within a given timeframe, is a critical metric in evaluating the efficiency and capacity of TO strategies in next-generation 6G-enabled edge-cloud environments as shown in Figure 5. Throughput analysis was performed across simulation intervals of 100 to 200 seconds in the proposed hybrid DRL and HDGC architecture, and the results were compared with the baseline deep reinforcement learning with transfer learning (DRLT) method.

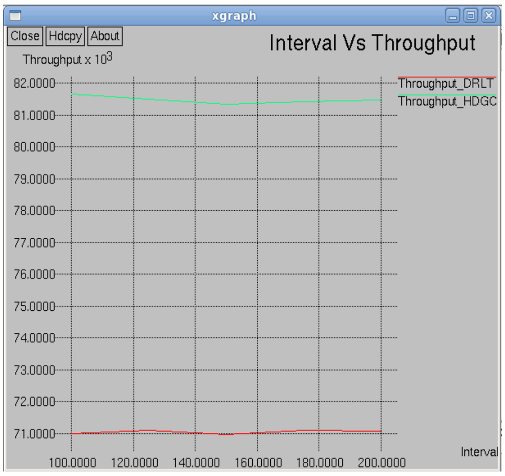

Figure 5. Interval vs throughput

5.3. Delay

Delay, or end-to-end latency, is a critical performance parameter in 6G-enabled edge-cloud networks, particularly for time-sensitive applications such as autonomous systems, remote healthcare, and industrial automation as shown in Figure 6.

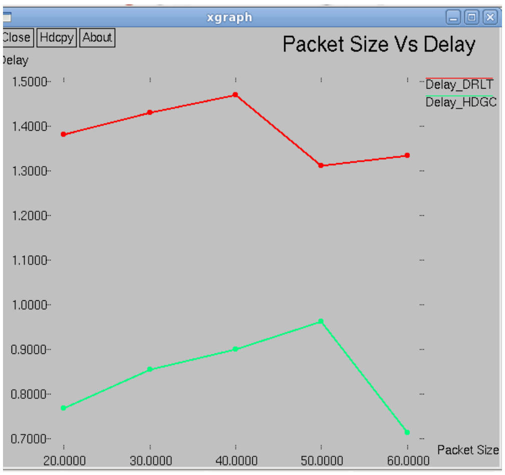

Figure 6. Packet size vs delay

5.4. Packet dropped

Packet dropped is a critical metric in evaluating the robustness and reliability of data transmission across edge-cloud networks, particularly in high-density, high-velocity 6G environments as shown in Figure 7. On average, HDGC reduces packet drops by more than 75%, showcasing its superior network resilience and adaptability under varying traffic loads. The pronounced performance advantage of HDGC stems from its hybrid architecture that combines real-time RL with adaptive heuristic search strategies.

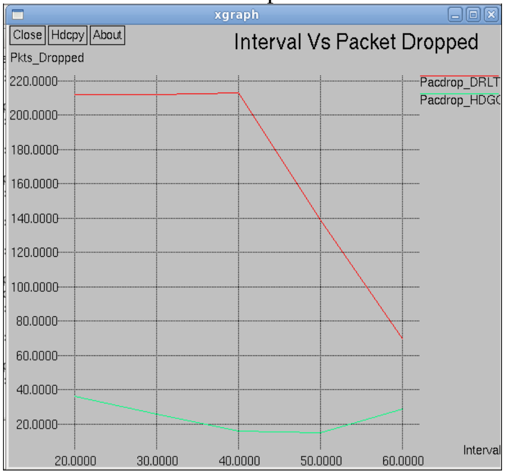

Figure 7. Packet size vs packet dropped

5.5. Data rate

Data rate is a pivotal performance metric in evaluating the capability of TO models within edge-cloud architectures, especially in 6G-enabled environments where ultra-high-speed and low-latency communication is imperative as shown in Figure 8. The observed performance improvements are rooted in HDGC's hybrid optimization design, which strategically blends DRL with genetic heuristics for dynamic task distribution and bandwidth allocation. The DRL agent, enhanced through attention-based architectures such as transformers and supported by transfer learning, adapts in real time to traffic fluctuations and device mobility.

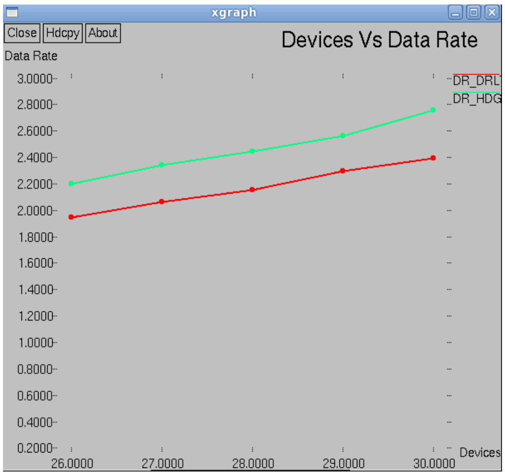

Figure 8. Packet size vs data rate

5.6. Average energy consumption

As seen in Table 1, energy economy is still a key design goal in 6G-enabled edge computing, where extended operation of battery-constrained IoT devices and edge nodes must be guaranteed without compromising computational performance. This trend persisted consistently; even at larger packet sizes, such as 50 and 60 bytes, HDGC maintained significantly lower energy consumption levels of 0.0232 Joules and 0.0299 Joules, respectively, compared to 0.0937 Joules and 0.0883 Joules consumed by DRLT. Using the RL policy with the help of the heuristic genetic optimization method supports to reduce the unnecessary task mobility and duplicate dispensation which usually results in the wastage of the energy.

Table 1. Values for average energy consumption

Packet size	DRLT	HDGC
20	0.0873672	0.0285472
30	0.0810113	0.0253235
40	0.0777652	0.02355
50	0.0936866	0.02319
60	0.0882806	0.02993

6. CONCLUSION

In this work, we proposed a hybrid TO framework that integrates DRL with GA to optimize task allocation in 6G-enabled edge–cloud environments. The framework enables intelligent decision-making that dynamically adapts to task availability, resource performance, and network topologies with real time modifications, which frame the nature and dynamic networks. Through this hybrid model, traditional Offloading methods are utilized for reliability, scalability, and effectiveness helps to achieve the optimal offloading decisions. The proposed work has a framework of incorporated GNN and transformer based attention, which produces enhancement in the overall performance of multimode distributed system with long term decision making and task dependency. The framework ensures data integrity and authenticity of the data with sensitive computation and communication among edge devices, servers, and cloud systems. The comparison between the proposed HDGC model and the baseline DRLT approach confirms the superiority of the hybrid solution across all evaluation metrics. HDGC achieves higher PDR, faster convergence, and improved throughput over increasing simulation times. Packet-drop comparisons indicate over a 75% reduction with HDGC, while data-rate comparisons show consistently higher performance as device numbers grow. Furthermore, energy-consumption comparisons highlight HDGC’s significant efficiency advantage, with far lower energy usage even at larger packet sizes. Overall, the comparison underscores the superior reliability, efficiency, and scalability of the proposed HDGC-based framework compared to DRLT.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kaniezhil Radhakrishnan	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Mong-Fong Horng		✓	✓	✓	✓	✓		✓	✓	✓			✓	
Siva Shankar Subramanian		✓	✓	✓	✓	✓		✓	✓	✓			✓	
Chun-Chih Lo		✓	✓	✓	✓	✓		✓	✓	✓			✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] L. Guo, J. Lin, X. Xu, and P. Li, “Algorithmics and complexity of cost-driven task offloading with submodular optimization in edge-cloud environments,” *arXiv preprint*, 2024, doi: 10.48550/arXiv.2411.15687.

[2] Z. Li, X. Zhou, Y. Liu, C. Fan, and W. Wang, “A computation offloading model over collaborative cloud-edge networks with optimal transport theory,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China, 2020, pp. 1006–1011, doi: 10.1109/TrustCom50675.2020.00134.

[3] A. Suzuki, M. Kobayashi, and E. Oki, “Multi-Agent Deep Reinforcement Learning for Cooperative Computing Offloading and Route Optimization in Multi Cloud-Edge Networks,” *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4416–4434, 2023, doi: 10.1109/TNSM.2023.3267809.

[4] M. R. Hossen and M. A. Islam, “Mobile Task Offloading under Unreliable Edge Performance,” *Performance Evaluation Review*, vol. 48, no. 4, pp. 29–32, 2021, doi: 10.1145/3466826.3466838.

[5] Y. Qin, J. Chen, L. Jin, R. Yao, and Z. Gong, “Task offloading optimization in mobile edge computing based on a deep reinforcement learning algorithm using density clustering and ensemble learning,” *Scientific Reports*, vol. 15, no. 1, pp. 1–25, 2025, doi: 10.1038/s41598-024-84038-3.

[6] S. Chakraborty and K. Mazumdar, “Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1552–1568, 2022, doi: 10.1016/j.jksuci.2022.02.014.

[7] Ş. Şentürk, V. Garousi, and N. Yumuşak, “A case study of gray-box fuzzing with byte- and tree-level mutation strategies in XML-based applications for exposing security vulnerabilities,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 33, no. 2, pp. 86–105, 2025, doi: 10.55730/1300-0632.4116.

[8] 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*, vol. 12, no. 1, pp. 1–15, 2023, doi: 10.1186/s13677-023-00461-3.

[9] R. Latip, J. Aminu, Z. M. Hanafi, S. Kamarudin, and D. Gabi, “Metaheuristic task offloading approaches for minimization of energy consumption on edge computing: a systematic review,” *Discover Internet of Things*, vol. 4, no. 1, pp. 1–30, 2024, doi: 10.1007/s43926-024-00089-y.

[10] B. Yang, X. Cao, J. Bassey, X. Li, T. Kroecker, and L. Qian, “Computation Offloading in Multi-Access Edge Computing Networks: A Multi-Task Learning Approach,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761212.

[11] X. Jin, S. Zhang, Y. Ding, and Z. Wang, “Task offloading for multi-server edge computing in industrial Internet with joint load balance and fuzzy security,” *Scientific Reports*, vol. 14, no. 1, pp. 1–26, 2024, doi: 10.1038/s41598-024-79464-2.

[12] G. Pandiyan and E. Sasikala, “Modelling Mobile-X Architecture for Offloading in Mobile Edge Computing,” *Intelligent Automation and Soft Computing*, vol. 36, no. 1, pp. 617–632, 2023, doi: 10.32604/iasc.2023.029337.

[13] X. Ma, K. Zong, and A. Rezaeipanah, “Auto-scaling and computation offloading in edge/cloud computing: a fuzzy Q-learning-based approach,” *Wireless Networks*, vol. 30, no. 2, pp. 637–648, 2024, doi: 10.1007/s11276-023-03486-3.

[14] S. Zhou, W. Jadoon, and I. A. Khan, “Computing Offloading Strategy in Mobile Edge Computing Environment: A Comparison between Adopted Frameworks, Challenges, and Future Directions,” *Electronics*, vol. 12, no. 11, pp. 1–30, 2023, doi: 10.3390/electronics12112452.

[15] H. B. Alla, S. B. Alla, and A. Ezzati, “DP-ARTS: Dynamic Prioritization for Adaptive Resource Allocation and Task Scheduling in Cloud Computing,” *IAENG International Journal of Computer Science*, vol. 52, no. 3, pp. 793–807, 2025.

[16] F. Saeik *et al.*, “Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions,” *Computer Networks*, vol. 195, 2021, doi: 10.1016/j.comnet.2021.108177.

[17] H. Tran-Dang and D. S. Kim, “FRATO: Fog Resource Based Adaptive Task Offloading for Delay-Minimizing IoT Service Provisioning,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 10, pp. 2491–2508, 2021, doi: 10.1109/TPDS.2021.3067654.

[18] H. Chen, W. Qin, and L. Wang, “Task partitioning and offloading in IoT cloud-edge collaborative computing framework: a survey,” *Journal of Cloud Computing*, vol. 11, no. 1, pp. 1–19, 2022, doi: 10.1186/s13677-022-00365-8.

[19] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, “Edge cloud offloading algorithms: Issues, methods, and perspectives,” *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–23, 2020, doi: 10.1145/3284387.

[20] M. Asim, Y. Wang, K. Wang, and P. Q. Huang, “A Review on Computational Intelligence Techniques in Cloud and Edge Computing,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 6, pp. 742–763, 2020, doi: 10.1109/TETCI.2020.3007905.

[21] R. Jindal, N. Kumar, and H. Nirwan, “MTFCT: A task offloading approach for fog computing and cloud computing,” in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2020, pp. 145–149, doi: 10.1109/Confluence47617.2020.9058209.

[22] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, “Toward Computation Offloading in Edge Computing: A Survey,” *IEEE Access*, vol. 7, pp. 131543–131558, 2019, doi: 10.1109/ACCESS.2019.2938660.

[23] S. Zhang, N. Yi, and Y. Ma, “A Survey of Computation Offloading with Task Types,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 8313–8333, 2024, doi: 10.1109/TITS.2024.3410896.

[24] M. Aazam, S. U. Islam, S. T. Lone, and A. Abbas, “Cloud of Things (CoT): Cloud-Fog-IoT Task Offloading for Sustainable Internet of Things,” *IEEE Transactions on Sustainable Computing*, vol. 7, no. 1, pp. 87–98, 2022, doi: 10.1109/TSUSC.2020.3028615.

[25] N. Kumari, A. Yadav, and P. K. Jana, “Task offloading in fog computing: A survey of algorithms and optimization techniques,” *Computer Networks*, vol. 214, 2022, doi: 10.1016/j.comnet.2022.109137.

[26] Y. Liu and Z. Yao, “Deep Learning-Based Multidimensional Assessment for Network Security Situations,” *IAENG International Journal of Computer Science*, vol. 52, no. 9, pp. 3056–3066, 2025.

[27] P. Sowmya, Vasudeva, and M. G. Rao, “Enhancing Accuracy of Chatbot with Optimizers and Deep Learning Algorithms,” *IAENG International Journal of Computer Science*, vol. 52, no. 9, pp. 3132–3140, 2025.

BIOGRAPHIES OF AUTHORS

Dr. Kaniezhil Radhakrishnan received her MCA degree from the Periyar University in the year 2001. She received her M.Phil. and Ph.D. Degrees from Annamalai university and Periyar University in 2007 and 2014, respectively. She is currently pursuing the Post Doctoral degree from National Kaohsiung University of Science and Technology, Taiwan. Her research interests include mobile computing, networking, spectral estimation, cognitive radio, and wireless communication. She can be contacted at email: kaniezhil@yahoo.co.in.

Prof. Dr. Mong-Fong Horng received his B.S. and M.S. degrees in Control Engineering from National Chiao Tung University in 1989 and 1991, respectively, and his Ph.D. degree in Computer Science from National Cheng Kung University, Taiwan, in 2003. He is a professor with the Department of Electronics Engineering, National Kaohsiung University of Science and Technology, Taiwan. He has served as the President of the Taiwanese Association of Consumer Electronics (TACE) and as Chair of the Tainan Chapter of the IEEE Signal Processing Society. His research interests include the internet of things, machine learning, computer networks, and medical informatics, along with related industrial collaborations. He can be contacted at email: mfhorng@nkust.edu.tw or mfhorng@ieee.org.

Dr. Siva Shankar Subramanian is currently working as an Professor and Head IPR in the Department of Computer Science and Engineering, KG Reddy College of Engineering and Technology, Hyderabad, Telangana, India. He completed his B.Tech. in Anna University, M.Tech. in MS University and Ph.D. in Bharath University. He has completed his Post doc in IUH, Vietnam. He has published 20+ journal papers and 20+ patents. His research interest includes security, image processing, mobile computing, and networks. He can be contacted at email: drsivashankars@gmail.com.

Chun-Chih Lo received his M.S. degree in Computer Science from Tunghai University, Taiwan, in 2007 and his Ph.D. degree in Computer Science and Information Engineering from National Cheng Kung University in 2016. Since 2017, he has served as a postdoctoral research fellow at National Cheng Kung University and later at National Kaohsiung University of Science and Technology. He is currently an Assistant Research Fellow in the Department of Electronic Engineering at National Kaohsiung University of Science and Technology, Taiwan. His current research interests include information security, network technologies and applications, emerging areas in artificial intelligence, mobile networks and computing, cloud computing, and machine learning. He can be contacted at email: georgelo@nkust.edu.tw.