GMR
The Original

DEEP LEARNING-BASED TASK OFFLOADING AND SCHEDULING ALGORITHMS IN MOBILE EDGE COMPUTING: A COMPREHENSIVE REVIEW

Chetana Hasmukh Jain[1], Tanuja Satish Dhope (Shendkar)*[2]

[1*]Research Scholar (Electronics Engineering), Department of Electronics Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, Maharashtra, India. Email: kankariya0209@gmail.com,
Orchid Id: 0009-0003-1362-113X
[2] Department of Electronics and Communication Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, Maharashtra, India. Email: tanuja_dhope@yahoo.com, Orchid Id: 0000-0003-2907-8509
Corresponding Author :Tanuja Satish Dhope(Shendkar) tanuja_dhope@yahoo.com

ABSTRACT:
The introduction of IoT, 5G, and latency-sensitive applications has created a huge demand for efficient computing and resource management. Mobile edge computing (MEC) has been suggested as a viable framework by which cloud computing functionalities can be extended to the edge of the network and improve the QoS by reducing latency. Task offloading and scheduling are two very important problems in MEC because of varying network states, availability of resources, and the demands posed by the application. Traditional optimization methods are efficient, but they exhibit very high computational complexity and lack real-time adaptivity. The development of advanced deep learning approaches has led to intelligent and adaptive methods for addressing task offloading and resource scheduling in MEC. Deep learning includes several methods, such as Deep Neural Networks (DNN), Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), LSTM networks, and DRL. They exhibit very good optimization capabilities for latency, energy efficiency, execution time, and resource utilization. This paper provides a thorough review of the use of deep learning for task offloading and scheduling in Mobile Edge Computing . The different methods are described and categorized based on architecture and optimization criteria among other factors. Also, performance evaluation and future direction of research are covered. The survey will help researchers understand more about the intelligent MEC paradigm and how future wireless technologies can benefit from it.

Keywords: Mobile Edge Computing; Task Offloading; Scheduling Algorithms; Deep Learning; Deep Reinforcement Learning; Resource Allocation; IoT; 5G Networks; Edge Intelligence; Latency Optimization.

1) INTRODUCTION:

The proliferation of the Internet of Things (IoT), smart mobile applications, autonomy-based solutions, and time-critical services has raised unprecedented demands on conventional cloud computing infrastructures. Applications such as augmented reality,intelligent transportation, health monitoring, industrial automation, and video analytics need low-latency, highly reliable, and efficient processing that the conventional centralised architecture finds difficult to meet. Mobile Edge Computing (MEC) is a potential solution where in the computation and storage capabilities move nearer to the users, thus resulting in better Quality of Service (QoS) [12], [16]. Through MEC, mobile users and IoT entities can perform time-consuming operations at the nearby edge servers rather than locally or remotely in far-off clouds. The advantages here include reduced latency, power savings, and more effective use of resources [10], [13]. Furthermore, with the convergence of edge computing and fifth-generation (5G) networks, the deployment of highly intelligent and time-critical services is being greatly facilitated [6], [19]. Although there are many benefits in MEC task offloading and scheduling, it is still difficult to solve this problem due to dynamic conditions, device heterogeneity, differences in processing needs, insufficient edge computing resources, mobile users, and random nature of wireless links. Commonly, optimization strategies and heuristic algorithms face issues related to excessive computational load and low adaptiveness to the dynamically changing environment [14], [17]. As a result, recent efforts have been dedicated to the use of Artificial Intelligence (AI) and machine learning methods that allow achieving intelligent behavior and autonomous resource management [18]. In the field of AI, deep learning has proven to be a very successful approach that can help to overcome the existing difficulties in complex optimization tasks. With deep learning networks, it is possible to learn the nonlinear relationships between system variables and apply the optimal strategies according to historical and real-time data. Among other things, Deep Reinforcement Learning techniques such as DQN, DDPG, PPO, A3C, and SAC have shown great results in such tasks as computation offloading, resource management, service placement, and scheduling [2], [9]. Recent developments have made clear the significance of deploying deep learning capabilities at the network edge even more. The paper by McClellan et al. [6] focused on the opportunities to utilize deep learning technologies in 5G MECs, while the paper by Wang et al. [19] was focused on the idea of In-Edge AI where intelligence is embedded in communication,

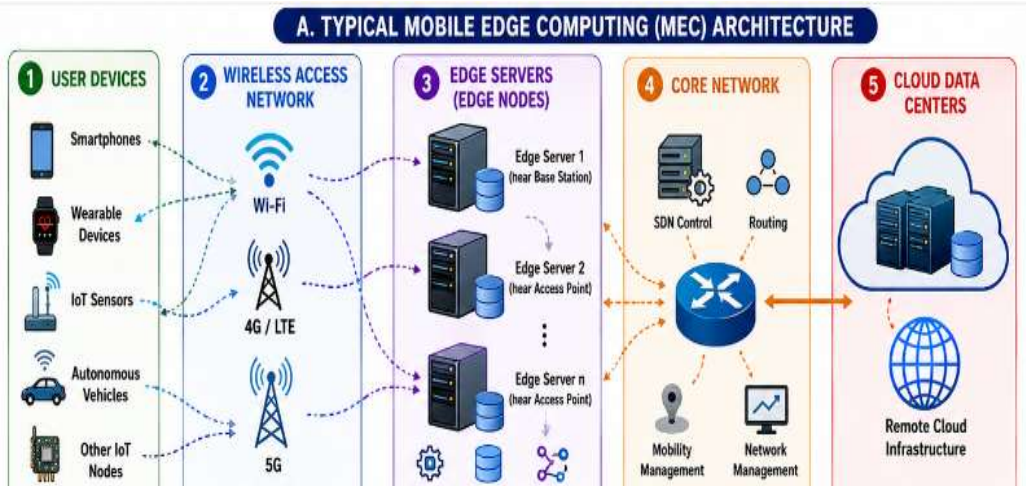
caching, and computing. In their study, Mahmoud et al. [8] provided information on deep learning models for cloud, fog, edge, and IoT computing that could enable the intelligent nature of distributed computing systems. Similarly, Xu et al. [7] described the importance of edge deep learning technology in computer vision and healthcare. Machine learning-based strategies for energy efficiency and computation offloading have been considered by many researchers. For instance, Ale et al. [2] discussed the energy-efficient computation offloading strategy based on delay-aware deep reinforcement learning and showed remarkable performance results regarding minimizing delay and energy efficiency. Mokgethi et al. [4] discussed the recent advancements in machine learning-based energy optimization and identified potential applications. Additionally, Lai et al. [5] examined lightweight deep learning models to be used in edge environments.

Task scheduling and resource allocation are also of equal importance in achieving high efficiency in MEC systems. Sardellitti et al. [13] explored the joint optimization of communication and computation resources, whereas Bi and Zhang [17] paid attention to computation rate maximization of wireless-powered MEC systems. Guo et al. [16] analyzed task offloading strategies in ultra-dense networks, and Sun et al. [20] explored energy-efficient mobility management approaches to tackle mobility problems. In summary, all these studies illustrate the rising demand for effective and dynamic scheduling mechanisms in edge networks.

Despite a large body of reviews concerning MEC systems architectures and communication considerations [12], MEC caching mechanisms [1], and machine learning techniques [18], there seems to be a lack of a survey that particularly discusses deep learning-enabled task offloading and scheduling algorithms. In particular, most of the current studies focus on algorithms or specific application scenarios but rarely discuss the architecture, strengths, weaknesses, and performances of the algorithms [3], [4]. The quick development of reinforcement learning algorithms and lightweight neural networks also requires an updated overview.

Thus, this paper provides a thorough overview of task offloading and scheduling approaches based on deep learning for Mobile Edge Computing.

This survey covers the architectural setup of MEC systems, deep learning models used for making decisions related to offloading and scheduling, performance measures, goals of optimization, challenges, and future research directions. Moreover, this survey makes comparative studies between the popular offloading and scheduling algorithms like DQN, DDPG, PPO, A3C, SAC, Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). The intention of this survey paper is to offer a comprehensive insight to the recent advancements in MEC technology .This would help in designing new generation MEC enabled mobile networks. In a typical MEC system architecture, user devices refer to smartphones , wearables, sensor devices, self-driving cars, and IoT devices.These devices generate computational tasks and data that may either be processed locally or offloaded to edge servers .The wireless access network provides communication between user devices and edge servers through technologies such as Wi-Fi, 4G LTE, and 5G networks. Efficient communication significantly affects system latency and energy consumption. Edge servers are deployed near base stations or access points and provide computational and storage resources. These servers execute offloaded tasks with minimal delay and support real-time applications .The core network interconnects edge nodes and provides routing and management functionalities. It facilitates communication between edge servers and remote cloud infrastructures .Cloud data centers offer abundant computational resources and storage capabilities. Tasks requiring extensive processing power may be forwarded from edge servers to cloud servers when necessary.



2 . LITERATURE SURVEY

The survey of related works is presented in Table 1below.

TABLE1 : SURVEY OF LITERATURE WORKS

Sr. No	Authors	Year	Focus Area	Key Contribution	Research Gap	Methodology
1	Djigal, H., Yu,	2022	Allocating resources in edge computing with machine learning	examines methods for machine learning and deep learning.	survey-based work; neither real-time assessment of performance	Joint optimization of offloading and resource allocation
2	M. Huang, Q. Zhao, Y. Chen, S. Feng, and F. Shu.	2021	The Multi-Objective Whale Optimization Algorithm (MOWOA) operates in mobile edge computing to offload computations.	Enhancing the practicability of algorithms in real-world edge computing	The Multi-Objective Whale Optimization Algorithm (MOWOA) operates in mobile edge computing to offload computations.	Improving the performance of the MOWOA algorithm in complex communication and offloading environments.
3	Z. Wan,D.Xu,I.Ahmad	2021	The swap matching algorithm is employed	optimizing MEC server and channel allocation.	Heuristic algorithm for optimizing MEC server allocation and resource management to minimize overall task delay.	Focus on minimizing energy consumption while considering task execution latency.
4	M.Tang,V.W.S.Wong	2022	Deep reinforcement learning distributed algorithm	The algorithm lowers the average delay to dropped task ratio,	The algorithm lowers the ratio of average delay to dropped tasks, particularly for tasks.	Model-free deep reinforcement learning-based distributed algorithm.
5	A.Shakarami,M.Ghobaei-Arani,A.Shahidnejad 29	2020	environments that rely on machine learning.	Important research questions remain unresolved in ML-based offloading mechanisms.	mechanisms into reinforcement, supervised, and unsupervised learning-based approaches.	partitioning, security, fault tolerance, mobility, scalability, scheduling, and interoperability in MEC environments.

6	Z,Ali,L.Jiao,TBaker,G.Abhas,Z,H.Abhas, S.Khaf 30	2022	Energy-efficient deep learning-based offloading scheme (EEDOS).	Not handles applications with non-linear call graphs or parallel executions.	EEDOS scheme outperforms other methods in offloading accuracy by considering energy consumption in its model.	Explore multi-user scenarios for generating training datasets for the deep neural network (DNN).
7	Lu et al. Electronics,	2024	Reduced energy consumption (~20%) while keeping latency within limits	Energy-efficient scheduling with load balancing	Limited to static server assumptions; does not address mobility or dynamic service demand	Stage 1: server activation decisions; Stage 2: load balancing decisions controlled with DRL
8	Qayyum et al.	2025	Enhanced task success rate and throughput under high mobility	Improved task success in VANETs	Hybrid AI increases computational cost;	Combined supervised learning (prediction), reinforcement learning (adaptation), and particle swarm optimization (
9	Zhao et al.	2023	Reduced latency (12-30%) and achieved 2.5%faster training for dependent tasks	Reduced latency & faster training for dependent tasks	Transformer adds training overhead; scalability for resource-constrained devices not explored	Used Transformer encoder to capture task dependency graphs,
10	T. S. Dhope, T. Dikshit, U. Gupta, and K. Kartik .	2024	Computational task offloading with Deep Q-Learning (DQL) in MEC.	Introduces a reinforcement learning	Limited scalability analysis; does not fully explore heterogeneous	Implemented a **DQL model** that takes system states (task size, CPU cycles, channel conditions
11	Shital Langote, Tanuja S. Dhope	2022	Task offloading in MEC using deep learning models for improving execution efficiency.	Proposed deep learning-based framework that predicts task execution cost and	Lacks reinforcement learning integration for adaptive offloading; static deep learning approach	Developed a deep neural network (DNN) to learn task features and network conditions;

				decides offloading policy; .		

3. COMPARATIVE ANALYSIS OF DIFFERENT TASK OFFLOADING MODELS

Task Offloading Models-
Task offloading is a fundamental mechanism in Mobile Edge Computing (MEC) that enables resource-constrained devices to transfer computational tasks to nearby edge servers. The objective of task offloading is to reduce execution time, minimize energy consumption, and improve overall Quality of Service (QoS). Depending on the application requirements and network conditions, task offloading can be classified into three categories.

3.1 Full Offloading
In full offloading, the entire computational task generated by a mobile device is transferred to the edge server for execution. After processing, the results are transmitted back to the user device.that Reduces computational burden on mobile devices, Saves battery energy. Suitable for computationally intensive applications.

3.2 Partial Offloading
Partial offloading divides an application into multiple components. Some tasks are executed locally while others are offloaded to edge servers. that gives Better resource utilization . Lower latency compared with full offloading. and Enhanced flexibility Partial offloading is widely used in multimedia applications and IoT systems where different subtasks have varying computational requirements.

3.3 Dynamic Offloading
Dynamic offloading adapts task execution decisions according to current network conditions, computational resources, and energy constraints. Machine learning and deep reinforcement learning techniques have become increasingly popular for implementing dynamic offloading mechanisms.

3.4 Binary Offloading and Fine-Grained Offloading
Binary offloading either executes the entire task locally or completely offloads it to edge servers. Fine-grained offloading divides applications into smaller modules, allowing selective offloading of computational components. Fine-grained offloading provides higher flexibility and better performance but introduces additional complexity in dependency management.

3.4 Block Diagram
Once the task is offloaded, it is then sent to the edge server of the MEC network for processing. Task scheduling is done by the edge server in order to effectively assign computational resources to several users and applications. In order to optimize the resource allocation through better scheduling, deep learning and deep reinforcement learning techniques are used. These intelligent techniques learn from the system environment and adapt accordingly for the execution of the tasks.

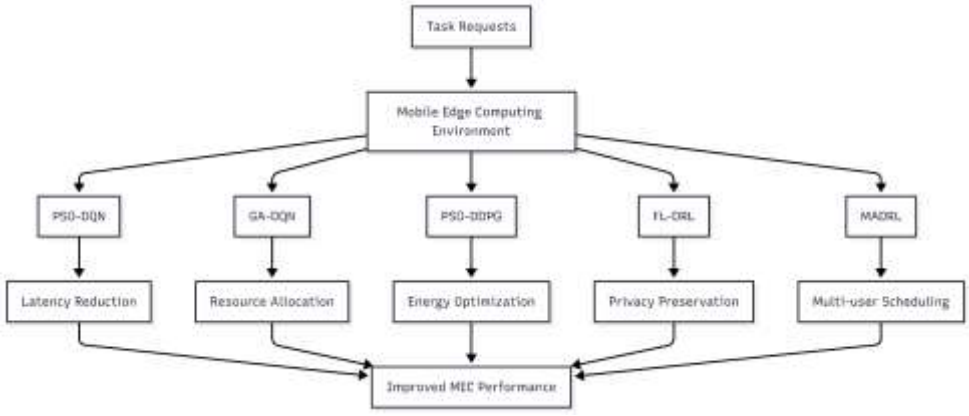


Fig.3.4. Block diagram of deep learning-based task offloading and scheduling in Mobile Edge Computing (MEC)

Through proper task scheduling, resources will be optimally assigned. Consequently, low latency, minimal energy consumption, and shorter execution time along with QoS are all achieved. This intelligent MEC network is capable of supporting real-time application in IoT, 5G, smart cities, health care, and autonomous systems.

3.5 Flowchart

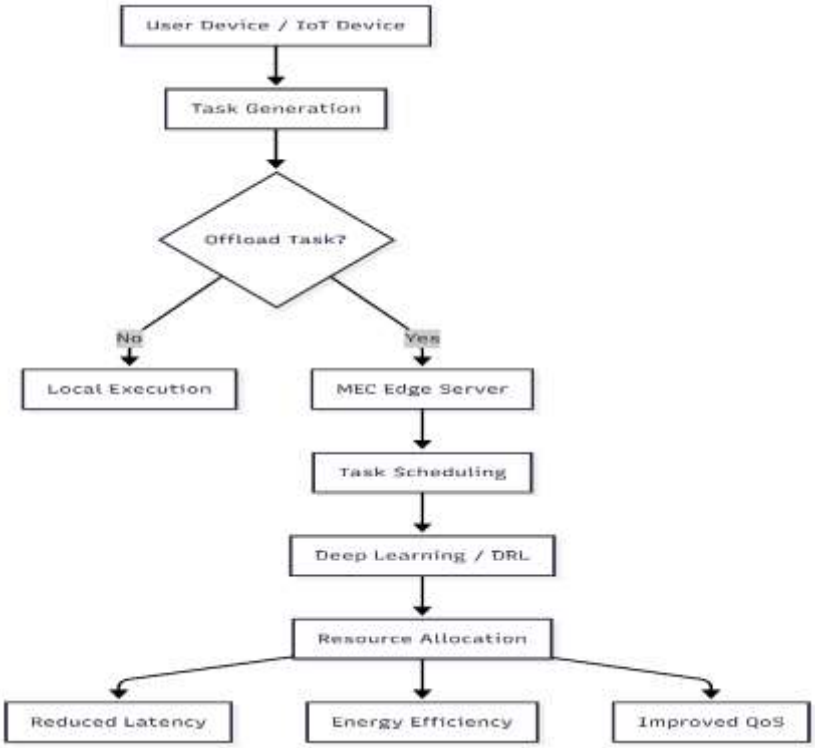
The flow diagram starts from the stage at which the computational task comes out of the user device or an IoT device. After that, the computation task is forwarded to the task offloading block. The computation task is determined to run on the device or be offloaded to the MEC server.

Fig.3.5. Flowchart of Deep Learning-Based Task Offloading and Scheduling in Mobile Edge Computing

If the device can carry out the computation task, then it will run on the device; otherwise, it will be offloaded to the MEC server. In the MEC server, there is task scheduling and resource allocation carried out using deep learning and deep reinforcement learning algorithms.

4. SCHEDULING ALGORITHMS IN MOBILE EDGE COMPUTING

Task scheduling determines how computational resources are allocated among multiple users and applications. Efficient scheduling algorithms are essential for reducing latency, improving throughput, and maximizing resource utilization



4.1 First Come First Serve (FCFS)

- FCFS executes tasks according to their arrival order. Simple implementation . Low computational complexity.

4.2 Round Robin Scheduling

- Round Robin allocates equal CPU time slices to each task. Fair resource allocation.

Suitable for time-sharing systems.

4.3 Priority-Based Scheduling

- Tasks are assigned priorities according to urgency and resource requirements. Improved response time for critical applications. Better Quality of Service.

Priority scheduling is extremely adopted in healthcare and industrial automation applications

4.4 Heuristic Scheduling Algorithms

Particle Swarm Optimization (PSO)

PSO mimics the collective behavior of birds and fish to optimize task allocation.it has Fast convergence. Easy implementation having limitation as Susceptible to local optima.

Ant Colony Optimization (ACO)

ACO is based on the pheromone communication behavior of ants. which has Efficient path optimization. Robust performance having limitation as High computational overhead.

4.5 Reinforcement Learning-Based Scheduling

Reinforcement learning enables agents to learn optimal scheduling policies through interactions with the environment having Adaptive decision-making.that Suitable for dynamic environments.Limitation as Requires extensive training. High computational cost.

5. DEEP LEARNING METHODS FOR TASK OFFLOADING AND SCHEDULING

Deep learning has emerged as a powerful approach for solving complex optimization problems in Mobile Edge Computing. Deep learning algorithms are capable of extracting hidden patterns from large datasets and making intelligent decisions under uncertain environments.

5.1 Deep Neural Networks (DNN)

Deep Neural Networks consist of multiple hidden layers that learn nonlinear relationships between input and output variables.

Working Principle

DNN models receive input parameters such as task size, channel state information , CPU frequency energy consumption and available bandwidth. The network processes these parameters and predicts optimal offloading decisions. having advantages as strong feature extraction capability. which can handle large datasets effectively and high prediction accuracy. Limitations as Large training datasets required. And High computational complexity. Applications can be included as resource allocation. task classification. latency prediction. Edge server selection.

5.2 Convolutional Neural Networks (CNN)

CNNs are specialized deep learning architectures designed for extracting spatial features.

Working Principle

CNN employs convolution and pooling layers to identify significant features from input data. Having advantages as automatic feature extraction. Reduced parameter complexity. High accuracy. Limitations as it requires significant computational resources. which is less suitable for sequential data. Applications in MEC can be image and video processing. traffic prediction. resource management.

5.3 Recurrent Neural Networks (RNN)

Recurrent Neural Networks are designed to process sequential and time-dependent data.

Working Principle

RNN utilizes feedback connections to maintain information from previous time steps.

5.4 Long Short-Term Memory Networks (LSTM)

LSTM networks are an improved version of RNN that overcome the vanishing gradient problem.

Working Principle

LSTM contains as input gate, forget gate output gate Memory cell. These components enable the network to preserve long-term information. Advantages as Excellent long-term memory capability. High prediction accuracy. Robust performance in dynamic environments. Limitations as Increased computational complexity. Longer training time.

Applications in MEC

- Workload prediction.
- Energy consumption estimation.
- Task scheduling.
- Mobility prediction.
- Resource allocation.

Model	Strengths	Weaknesses	Applications
DNN	High accuracy and feature extraction	Large training data required	Resource allocation
CNN	Efficient spatial feature learning	Computationally intensive	Image processing and traffic prediction
RNN	Suitable for sequential data	Vanishing gradient problem	Dynamic scheduling
LSTM	Captures long-term dependencies	High training complexity	Workload and mobility prediction

6. HYBRID OPTIMIZATION TECHNIQUES

Hybrid optimization strategies hold great importance in terms of enhancing task scheduling and resource allocation in Mobile Edge Computing (MEC). The hybrid strategy that utilizes PSO and DQN, which is referred to as PSO-DQN, optimizes task offloading. PSO-DQN leads to better efficiency in terms of rapid convergence, efficient resource allocation, minimal latency and energy consumption. The hybrid approach of integrating Genetic Algorithm (GA) with Deep Reinforcement Learning (DRL) aims at exploring the problem domain to avoid local optimum solutions and achieve better task completion. The combination of PSO and Deep Deterministic Policy Gradient (DDPG) helps tackle the challenge of continuous optimization, in addition to leading to faster policy convergence and minimized execution time. The optimization strategy relying on federated learning is designed to enable a group of edge devices collaborate during the training of machine learning algorithms without exchanging private information. Multi-agent Deep Reinforcement Learning (MADRL) is used to optimize task scheduling and resource allocation, thanks to its scalable, adaptable and efficient nature. The above techniques can be applied in various systems including IoT systems, healthcare systems, smart cities, autonomous driving vehicles, vehicular edge computing, and industrial IoT systems.

7.COMPARATIVE ANALYSIS OF DEEP LEARNING AND HYBRID ALGORITHM

A comparison of different optimization and learning techniques for task offloading in Mobile Edge Computing (MEC) reveals significant differences in their efficiency and reliability. Traditional approaches such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) provide acceptable performance, but GA generally experiences higher energy usage and moderate latency, whereas PSO achieves faster convergence, lower delay, and improved task completion. Deep learning methods, including Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) networks, enhance decision-making capabilities by learning patterns from system data. DNN offers reduced latency and good task completion performance, while LSTM further improves efficiency by effectively handling time-dependent information, resulting in lower energy consumption and higher completion rates. Deep reinforcement learning algorithms demonstrate even better performance. Deep Q-Network (DQN) minimizes delay and execution time while maintaining high task success, and Deep Deterministic Policy Gradient (DDPG) provides excellent energy efficiency and supports continuous optimization of offloading decisions. Among the policy-based approaches, Proximal Policy Optimization (PPO) achieves a strong balance between delay, energy consumption, execution speed, and task completion reliability**.** Soft Actor-Critic (SAC) delivers the most favorable overall results, providing superior latency reduction, efficient energy utilization, and the highest task completion capability. Consequently, advanced deep reinforcement learning methods, particularly SAC and PPO, are highly effective choices for resource allocation and task offloading in dynamic MEC environments due to their ability to optimize multiple performance metrics simultaneously as shown in table below.

Table 2: Comparison of Deep Learning Models

Algorithm	Learning Type	Strengths	Limitations	Applications
DNN	Supervised Learning	High accuracy	Large dataset required	Resource allocation
CNN	Supervised Learning	Feature extraction	High computation cost	Image processing
RNN	Sequential Learning	Handles time-series data	Vanishing gradient problem	Traffic prediction
LSTM	Sequential Learning	Long-term dependency	Complex architecture	Workload prediction
DQN	Reinforcement Learning	Adaptive decisions	Slow convergence	Task offloading
DDPG	Reinforcement Learning	Continuous action spaces	Training instability	Resource management
PPO	Reinforcement Learning	Stable learning	High computational complexity	Dynamic scheduling
SAC	Reinforcement Learning	Improved exploration	Increased training time	Energy optimization

Hybrid algorithms in Mobile Edge Computing (MEC) combine the strengths of optimization techniques and deep reinforcement learning methods to improve task offloading and resource management. PSO-DQN**,** which integrates Particle Swarm Optimization and Deep Q-Network, primarily focuses on reducing latency by utilizing the fast convergence capability of PSO and the adaptive learning ability of DQN, resulting in efficient and rapid decision-making. GA-DQN combines Genetic Algorithm with DQN to achieve effective resource allocation and enhance system performance by avoiding local optimum solutions through evolutionary search mechanisms. PSO-DDPG merges PSO with Deep Deterministic Policy Gradient to optimize energy consumption and provide efficient continuous control for dynamic resource allocation and power management. FL-DRL, which incorporates Federated Learning with Deep Reinforcement Learning, aims to preserve user privacy by enabling distributed learning without sharing raw data, thereby providing secure and intelligent decision-making across edge devices. MADRL (Multi-Agent Deep Reinforcement Learning) employs multiple cooperative agents to perform multi-user scheduling and resource allocation, offering

improved scalability and adaptability in large-scale and highly dynamic MEC environments. Overall, these hybrid approaches exploit the advantages of both optimization and learning-based methods to achieve better latency reduction, resource utilization, energy efficiency, privacy preservation, and scalability, making them promising solutions for next-generation Mobile Edge Computing systems shown below.

Table 3: Comparison of Hybrid Optimization Techniques

Hybrid Algorithm	Components	Major Objective	Advantages
PSO-DQN	PSO + DQN	Latency Reduction	Fast convergence
GA-DQN	GA + DQN	Resource Allocation	Avoids local optima
PSO-DDPG	PSO + DDPG	Energy Optimization	Efficient continuous control
FL-DRL	Federated Learning + DRL	Privacy Preservation	Distributed intelligence
MADRL	Multi-Agent DRL	Multi-user Scheduling	Scalability

The comparative analysis of various optimization and intelligent learning techniques for Mobile Edge Computing (MEC) highlights their effectiveness in terms of latency, energy consumption, execution time, and task completion ratio. Conventional optimization methods such as Genetic Algorithm (GA**)** and Particle Swarm Optimization (PSO) provide acceptable performance, with GA showing moderate delay and execution speed but requiring relatively higher energy, while PSO achieves faster convergence and improved task completion with lower latency. Deep learning approaches, including Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) networks, enhance system efficiency by learning complex patterns from data. DNN offers reduced delay and high task success, whereas LSTM delivers better results through its capability to capture temporal dependencies, leading to lower energy usage and higher completion rates. Deep reinforcement learning techniques further improve overall performance. Deep Q-Network (DQN) minimizes response time and computational overhead while maintaining a high success rate in task execution. Deep Deterministic Policy Gradient (DDPG) provides enhanced energy efficiency and supports continuous decision-making, resulting in improved resource utilization. Among policy-based algorithms, Proximal Policy Optimization (PPO) achieves a strong balance between low latency, minimal energy consumption, rapid execution, and reliable task completion. Soft Actor-Critic (SAC) demonstrates the most superior performance, offering excellent delay reduction, efficient energy management, and the highest task completion capability. Overall, the results indicate that advanced deep reinforcement learning methods outperform conventional optimization techniques, making them highly suitable for intelligent task offloading and resource management in dynamic MEC environments.

Table 4: Performance Comparison

Algorithm	**Latency**	**Energy Consumption**	**Execution Time**	**Task Completion Ratio**
GA	Medium	High	Medium	Medium
PSO	Low	Medium	Low	High
DNN	Low	Medium	Low	High
LSTM	Very Low	Low	Low	Very High
DQN	Very Low	Low	Very Low	High
DDPG	Low	Very Low	Low	Very High
PPO	Very Low	Very Low	Very Low	Excellent
SAC	Excellent	Excellent	Low	Excellent

CONCLUSION

MEC represents a promising concept that allows reducing latency in latency-sensitive applications by bringing computational resources closer to users. Proper task offloading and scheduling are important factors required for minimizing latency, energy usage, and ensuring high-quality service provision. Existing optimization techniques, such as heuristics and metaheuristics, have yielded acceptable results; however, the applicability of such techniques is constrained by the inability of these techniques to adapt to highly dynamical MEC systems. In recent years, deep learning techniques have dramatically changed the intelligent management of computing resources in MEC systems. Specifically, Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory Network (LSTM), and Deep Reinforcement Learning (DRL) techniques have been successfully applied to optimize task offloading and scheduling. Specifically, DRL techniques such as Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), Advantage Actor Critic (A2C), Proximal Policy Optimization (PPO), and Soft Actor Critic (SAC) have shown their excellence in handling uncertainties and dynamics. Hybrid approaches incorporating optimization and deep learning result in better performance through efficient exploitation and exploration. Hybrid approaches result in efficient utilization of resources, less delay time, and energy efficiency. There are various challenges faced by researchers while designing intelligent MEC, including scalability, security concerns, mobile network management, high computational cost, and requirement of real-time. Future trends in research include federated learning, digital twin, Explainable AI, multi-

agent reinforcement learning, green edge computing, and 6G edge intelligence. These technologies are expected to deliver reliable, scalable, and self-sustaining solutions for future-generation wireless networks. Thus, deep learning-based offloading and scheduling schemes will be crucial for the development of intelligent Mobile Edge Computing systems.

REFERENCES

[1] Y. Zhao and W. Zhang, "A Survey on Caching in Mobile Edge Computing," Wireless Communications and Mobile Computing, vol. 2021, pp. 1–20, 2021.

[2] L. Ale, N. Zhang, X. Fang, X. Chen, S. Wu, and L. Li, "Delay-Aware and Energy-Efficient Computation Offloading in Mobile Edge Computing Using Deep Reinforcement Learning," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9398–9410, 2021.

[3] S. Miriyala and V. R. Chirra, "Deep Learning Approaches for Computation Offloading in Edge Computing: A Critical Review," Telecommunication Systems, vol. 89, no. 1, pp. 1–25, 2026.

[4] C. Mokgethi, T. Sigwele, K. C. Bhende, and A. Maenge, "Machine Learning Centered Energy Optimization in Mobile Edge Computing: A Review," International Journal of Informatics and Communication Technology, vol. 15, no. 2, pp. 465–476, 2026.

[5] N. Lai, D. A. Dewi, S. S. Maidin, W. Xiao, and Q. Hu, "A Comprehensive Review of Lightweight Deep Learning Models for Edge Computing with Future Directions," Discover Computing, vol. 29, no. 110, pp. 1–35, 2026.

[6] M. McClellan, C. Cervelló-Pastor, and S. Sallent, "Deep Learning at the Mobile Edge: Opportunities for 5G Networks," Applied Sciences, vol. 10, no. 14, pp. 4735–4756, 2020.

[7] Y. Xu, T. M. Khan, Y. Song, and E. Meijering, "Edge Deep Learning in Computer Vision and Medical Diagnostics: A Comprehensive Survey," Artificial Intelligence Review, vol. 58, no. 93, pp. 1–45, 2025.

[8] M. M. E. Mahmoud et al., "Deep Learning Models for Cloud, Edge, Fog and IoT Computing Paradigms: Survey, Recent Advances and Future Directions," Computer Science Review, vol. 49, pp. 100568, 2023.

[9] S. Wang, R. Urgaonkar, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic Service Placement for Mobile Micro-Clouds with Predicted Future Costs," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 4, pp. 1002–1016, 2021.

[10] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient Multi-User Computation Offloading for Mobile Edge Computing," IEEE/ACM Transactions on Networking, vol. 24, no. 5, pp. 2795–2808, 2021.

[11] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical Fog-Cloud Computing for IoT Systems," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9893–9905, 2021.

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358.

[13] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile Edge Computing," IEEE Transactions on Signal and Information Processing over Networks, vol. 1, no. 2, pp. 89–103.

[14] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic Computation Offloading for Mobile Edge Computing with Energy Harvesting Devices," IEEE Journal on Selected Areas in Communications, vol. 34, no. 12, pp. 3590–3605.

[15] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease Prediction by Machine Learning over Big Data from Healthcare Communities," IEEE Access, vol. 5, pp. 8869–8879.

[16] H. Guo, J. Liu, and J. Zhang, "Computation Offloading for Multi-Access Mobile Edge Computing in Ultra-Dense Networks," IEEE Communications Magazine, vol. 56, no. 8, pp. 14–19.

[17] S. Bi and Y. J. Zhang, "Computation Rate Maximization for Wireless Powered Mobile Edge Computing," IEEE Transactions on Wireless Communications, vol. 17, no. 6, pp. 4177–4190.

[18] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine Learning for Wireless Networks with Artificial Intelligence," IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3039–3071.

[19] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication," IEEE Network, vol. 33, no. 5, pp. 156–165.

[20] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-Aware Mobility Management for Mobile Edge Computing," IEEE Transactions on Vehicular Technology, vol. 68, no. 8, pp. 7643–76