British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 7, Issue 3, 2024 (pp. 107-117)
www.abjournals.org

A REVIEW OF TASK OFFLOADING ALGORITHMS WITH DEEP REINFORCEMENT LEARNING

Ahmad Umar Labdo[1*], Ajay Singh Dhabariya[2], Zainab Mukhtar Sani[1], and Musa Abubakar Abbayero[1]

[1]Department of Computer Applications, Mewar University Gangrar, Chittorgarh, Rajasthan, India

[2]Assistant Prof. Department of Computer Applications, Mewar University Gangrar, Chittorgarh, Rajasthan, India.

[*]Corresponding Author's Email: mails2labdo@gmail.com

Cite this article:

Labdo A. U., Dhabariya A. S., Sani Z. M., Abbayero M. A. (2024), A Review of Task Offloading Algorithms with Deep Reinforcement Learning. British Journal of Computer, Networking and Information Technology 7(3), 107-117. DOI: 10.52589/BJCNIT-UGHJH8QG

Manuscript History

Received: 18 Jun 2024

Accepted: 17 Aug 2024

Published: 23 Aug 2024

ABSTRACT: *Enormous data generated by IoT devices are handled in processing and storage by edge computing, a paradigm that allows tasks to be processed outside host devices. Task offloading is the movement of tasks from IoT devices to an edge or cloud server –where resources and processing capabilities are abundant– for processing, it is an important aspect of edge computing. This paper reviewed some task-offloading algorithms and the techniques used by each algorithm. Existing algorithms focus on either latency, load, cost, energy or delay, the deep reinforcement phase of a task offloading algorithm automates and optimizes the offloading decision process, it trains agents and defines rewards. Latency-aware phase then proceeds to obtain the best offload destination in order to significantly reduce the latency.*

KEYWORDS: Edge, Fog, Round-trip, Cloud.

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 7, Issue 3, 2024 (pp. 107-117)

INTRODUCTION

The Internet of Things has become increasingly popular in recent years (IoT). IoT has greatly aided in the advancement of artificial intelligence (AI) by supplying enough data for model training and inference. In light of this context and trend, the conventional cloud computing model may still have significant difficulties handling the vast amounts of data produced by the Internet of Things on its own and satisfying the ensuing practical demands (Hua et al., 2023). Bello and Zeadally (2022) state that many hardware devices and sensors are being produced and used worldwide as a result of the Internet of Things (IoT). These hardware components, or sensors, are capable of perceiving the physical surroundings and converting that information into data. Data consumers can access cloud data based on their specific requirements and then extract the necessary information once these enormous amounts of data have been transferred to the cloud for processing or storage.

Since its inception, cloud computing has been widely used and has significantly altered peoples' lifestyles. Numerous major corporations have introduced their own cloud computing services, such as Google Cloud, Amazon Web Services, and Microsoft Azure (Jiang & He, 2023). IoT systems are widely used in many different industries, such as manufacturing, healthcare, agriculture, and smart cities (Nguyen et al., 2024). These systems still have drawbacks, though, including low automaticity, high latency, inefficient energy use, inefficient bandwidth utilization, and a lack of security. It has been suggested that edge computing and blockchain be incorporated into IoT to overcome these drawbacks. However, this integration is difficult and hasn't been thoroughly studied.

Thankfully, the problem is clarified by emerging edge computing, which significantly lowers latency and boosts efficiency by moving data processing from the remote network core to the local network edge. The review of edge computing and blockchain integration into the Internet of Things systems is the main goal of current research. Research is needed to address applications of blockchain-based edge potential usages while taking security requirements into account. Furthermore, there is a need for in-depth analyses of difficulties and perspectives on the future course of edge IoT systems based on blockchain technology (Nguyen et al., 2024).

Edge computing, according to Dikonimaki (2023), is the practice of situating consolidated data centers close to the user (as far as networks go) to minimize latency, offer scalability, and use less energy. Edge computing is a paradigm that is becoming increasingly important in the changing digital landscape. It is particularly important for real-time, low-latency applications and Internet of Things (IoT) environments. Notwithstanding its benefits, edge computing is severely limited in terms of storage capacity and is prone to reliability problems because of its resource constraints and exposure to harsh environments (Gao et al., 2024).

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 7, Issue 3, 2024 (pp. 107-117)

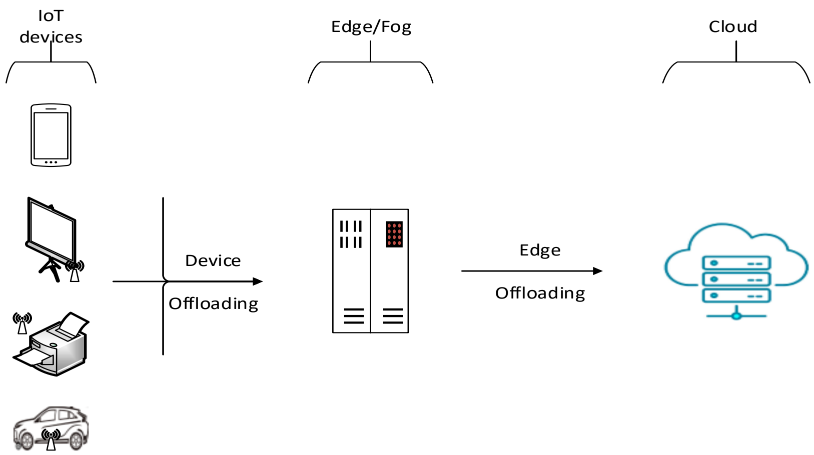

Fig. 1: Edge Computing Architecture

TASK OFFLOADING

Computational offloading, or the transfer of tasks from one device to another, is one of the main issues that must be resolved to enable edge computing to its full potential. Dikonimaki (2023) designs, develops and compares various offloading mechanisms in addition to examining the various factors that affect the decision to offload. The objective is to offer a dynamic and lightweight offloading decision policy to improve user performance and Quality of Experience (QoE).

Task offloading is moving computational workloads to a more potent remote server or cloud infrastructure from a device with limited resources, like a smartphone or Internet of Things device. By utilizing the greater processing power of distant servers, this can improve user experience, save energy, and improve performance. Task offloading is the practice of delegating computational tasks from local devices to remote servers to optimize resource use and performance (Wang et al., 2021).

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 7, Issue 3, 2024 (pp. 107-117)
www.abjournals.org

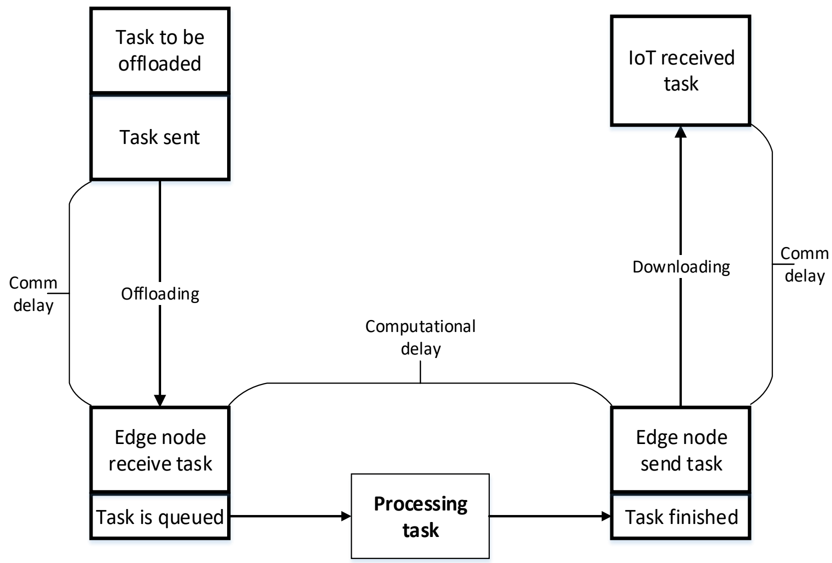

Fig. 2: Delay in Task Offloading

REVIEWED WORKS

Huang et al. (2017) state that modern cloud computing platforms are up against obstacles like the massive amount of data traffic generated by crowdsourcing and the intensive processing requirements of typical deep learning applications. Lately, there has been talk of Edge Computing as a useful method for cutting down on resource usage. In this paper, the idea of edge computing was introduced while proposing an edge learning framework, the research shows how effective the framework is at lowering running times and network traffic. In particular, Wang and Ma (2020) offer insights on how to leverage deep learning advances to facilitate edge applications from four domains: smart multimedia, smart transportation, smart city, and smart industry. In the article, they provide a thorough survey of the most recent efforts on deep learning-enabled edge computing applications.
They also highlight the most important and promising research directions and challenges that are present. This survey will stimulate additional studies and contributions in this exciting area.

Gao et al. (2024) present a customized storage mechanism for edge computing that prioritizes data reliability and space efficiency to address these issues. Relation factorization, column clustering, and erasure encoding with compression are the three main components of their approach. Through the deconstruction of intricate database tables and the optimization of data organization within these sub-tables, they were able to successfully lower the required storage space. Erasure encoding is another layer of reliability that was added. Extensive tests on TPC-H datasets validate the method, showing up to 38.35% storage savings and up to 3.96x time efficiency gains in some scenarios. The clustering method also indicates the possibility of up to 40.41% more storage savings.

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

An edge computing offloading algorithm for software-defined Internet of Things is presented by Zhu et al. (2023). First, a software-defined edge computing (SDEC) architecture is suggested to supply a global state for decision-making. Software-defined IoT incorporates the edge layer into the control layer, and via east-west message exchange, several controllers exchange the global network status data. Additionally, a deep reinforcement learning-based edge computing offloading algorithm for software-defined IoT (ECO-SDIoT) is introduced. It permits the controllers to transfer the computing task to the edge server that best suits the task requirements, global states, and reward. Ultimately, task processing energy consumption, task completion rate, load balancing of edge servers, and unit task processing latency were used as performance metrics to assess edge computing offloading.

Mobile devices can delegate some of their tasks to cloud resources situated close to them as edge servers. This will expedite the completion of tasks to satisfy the growing computational requirements of mobile applications. Decisions regarding offloading have been made using a variety of methods. In contrast to traditional reinforcement learning techniques, the RL approach presented in this paper takes into account delayed feedback from the environment. The outcomes of the simulation demonstrate that the suggested strategy greatly improved the traditional reinforcement techniques while effectively managing the environment's random delayed feedback (Aghasi & Rituraj, 2022).

Chakraborty and Mazumdar (2022), in their research, found that the threat of resource limitation posed by mobile devices (MDs) is subdued by Sensor Mobile Edge Computing (SMEC), which propels energy-constrained sensor tasks to edge servers. The majority of previous research exclusively addressed offloading concerns. In the path of MDs, a task might, however, depend on certain related tasks that were completed in the previous edge server. The collection of dependent data enables task execution. In a multiuser, multichannel setting, this study explains how to dynamically select an edge cloud for task offloading and verifies the dependencies of the tasks. To complete the execution, the suggested dynamic edge server selection for the inter-edge dependent task scheme gathers data from several allied edge nodes. To identify the ideal solution, this paper uses an optimization technique based on Genetic Algorithms (GA) in the SMEC environment (GAME) (Chakraborty & Mazumdar, 2022). The study's outcome is the evaluation of several task dependencies while maintaining energy consumption and computational delay within the permitted transmission latency range.

Dayong et al. (2024) offered a thorough and in-depth explanation of the algorithms and workings of the MEC network's multiple IoT task offloading mechanism. The primary issues resolved by the mechanism, technical categorization, assessment techniques, and supplementary parameters are taken out and examined for every paper. Potential and new researchers will find it easier to quickly grasp the range of IoT task offloading approaches in MEC and identify relevant research avenues with the aid of this review. In a paper, Huang et al. (2019) proposed a Deep-Q Network (DQN) based task offloading and resource allocation algorithm for the MEC, motivated by the ensuing need for appropriate resource allocations for computation offloading via MEC. To reduce the overall offloading cost in terms of energy, computation, and delay costs, the MEC system was specifically taken into consideration in which each mobile terminal has multiple tasks offloaded to the edge server. A joint task offloading decision and bandwidth allocation optimization was

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

designed. Even though the suggested optimization problem is essentially a mixed integer nonlinear programming problem, it was solved by taking advantage of a recently developed DQN method (Huang et al., 2019). Comprehensive numerical outcomes demonstrate that the near-optimal performance can be attained with the suggested DQN-based method.

Due to the resource-constrained nature of Internet of Things (IoT) devices, tasks can be offloaded from IoT devices to adjacent mobile edge computing (MEC) servers, which can reduce response times while simultaneously conserving energy. However, because of the MEC server's constrained processing power, assigning a task to the closest one might not be the best course of action. Therefore, it is crucial to optimize resource management and offloading decisions simultaneously, though this has not yet been done. The decision to offload a task here refers to where to do so, and the management of resources in an MEC server denotes the amount of computing power assigned to a task. Akhavan et al. (2022) proposed the deep reinforcement learning-based offloading decision and resource management (DECENT) algorithm, which uses the advantage actor-critic method to optimize the offloading decision and computing resource allocation for each arriving task in real-time so that the cumulative weighted response time can be minimized. This algorithm takes into account the waiting time of a task in the communication and computing queues (which are ignored by most of the existing works) as well as task priorities.

Baker et al. (2024) stated that a task-offloading strategy in mobile edge computing systems should strike a balance between dependability, efficiency, adaptability, and trust management. This strategy takes into account the dynamic and resource-limited nature of edge environments while maximizing resource utilization, enhancing user experience, and meeting application-specific requirements. Evaluation of edge node trust is also necessary because these systems are susceptible to various attacks and privacy violations when offloading tasks. Nevertheless, not all of these essential components are found in the offloading techniques that are currently in use. To address the complex problems related to offloading in mobile edge computing systems, this research proposes a novel approach called "EDITORS" (energy-aware dynamic task offloading method utilizing Deep Reinforcement Transfer Learning, or DRTL) in Software-Defined Network (SDN) enabled edge computing environments. EDITORS developed a task-offloading system that integrates trusted edge nodes and prioritizes energy efficiency, timeliness, reliability, adaptability, and superior quality of the task-offloading plan compared to current task-offloading methods (Baker et al., 2024). This approach makes use of DRTL agents at edge nodes, which speak with the SDN controller to determine the best offloading options depending on the state of the network and the availability of resources. Six extensive simulations are run, and the results demonstrate that the EDITORS greatly improves energy efficiency while maintaining low-latency task completion when compared to the five offloading techniques currently in use (DDLO, DROO, DMRO, DRL without TL and SDN, and DRL with SDN). EDITORS comprises trust assessment, LSTM-based trusted edge device prediction, and transfer learning-based device adaptation for newly added devices, unlike other task offloading methods that just concentrate on task offloading.

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

SUMMARY OF REVIEWED WORKS

S/N	Title	Methodology	Datasets & Parameters	Result	Weakness
1	Deep reinforcement learning for online latency-aware workload offloading in mobile edge computing	Actor critic method, deep reinforcement learning and latency-aware optimization	Synthetic Datasets: Task sizes, processing capabilities of mobile devices, edge servers, network bandwidth and latency metrics.	The average weighted response time incurred by DECENT increases, while still maintaining a low level (< 500ms). DECENT achieves better offloading decision and resource management in both light and heavy workload scenarios.	Lack of consideration for network propagation delay and network variability
2	Deep reinforcement learning-based edge computing offloading algorithm for software-defined IoT	Software-Defined Edge Computing (SDEC) Architecture, controllers, DRL algorithm and Optimization Objectives	Synthetic Datasets: Task sizes, processing capabilities, network bandwidth and latency metrics	The paper achieved the optimization goals of low energy consumption, low latency, and load balancing for edge computing offload policies in an environment where the network state and task demands are dynamically changing.	The study did not consider network latency fluctuations, edge server proximity and real-time latency adaptation.
3	A latency-aware power-efficient reinforcement learning approach for task offloading in multi-access edge networks	Reinforcement learning that considers delayed feedback, Power Efficiency and Latency Awareness and System Model	Synthetic Datasets: Number of mobile devices, task specifications, network parameters, channel bandwidth and device capabilities	This paper demonstrates that the proposed method has not only made more optimal decisions, but also has acted faster in reaching the optimal policy than conventional SARSA algorithms.	The study did not cover the application and the performance of ensemble and hybrid machine learning, and comparative analysis was not performed.
4	Sustainable task offloading	Genetic Algorithm (GA)	Synthetic Datasets:	The genetic algorithm reduced task	Does not consider the centralized cloud

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

ISSN: 2689-5315
Volume 7, Issue 3, 2024 (pp. 107-117)

	decision using genetic algorithm in sensor mobile edge computing	based optimization technique in the SMEC, Task Offloading Strategy and Resource Allocation	Network conditions, task loads, channel bit rate, propagation delay and resource availability	completion times and energy consumption compared to traditional task offloading methods. GAs allowed for a more balanced and efficient distribution of tasks across the available resources in the MEC network.	server, task execution and sequencing
5	QoS-SLA-aware adaptive genetic algorithm for multi-request offloading in integrated edge-cloud computing in Internet of vehicles	QoS-SLA integration, adaptive genetic Algorithm, convergence analysis and comparative performance analysis (Comparison with Baseline Methods)	Vehicle-Crowd Interaction (VCI) – DUT dataset: Traffic conditions, vehicle speeds, computational loads, crossover rate, mutation rate, population size and termination condition	The algorithm executes requests 1.04, 1.23, 1.05 and 9.41 times faster than the PSO, random offloading, ACC, and AEC approaches. It has the fastest convergence with the shortest total execution time, it also gives a higher total execution time.	To investigate QoS-SLA-aware priority-based partial offloading solutions and to reduce convergence time
6	A pipelining task offloading strategy via delay-aware multi-agent reinforcement learning in Cyber Twin-enabled 6G network	Delay-aware Markov Decision Process (MDP), Multi-Agent Reinforcement Learning (MARL), Gate Transformer-XL and Centralized Training with Decentralized Execution	Synthetic Datasets: Network state information, task characteristics and resource availability Size of the task, Transmission power	The performance of the pipelining task offloading strategy generated by the delay-aware MADRL algorithm outperforms the traditional MADRL algorithm in random delay Cyber Twin-enabled 6G networks. The strategy also performs better in terms of reaction time.	The study only focused on delay, not other key factors in offloading.
7	EDITORS: Energy-aware Dynamic Task Offloading using Deep	DRTL agents, task-based model, LSTM, Transfer Learning and	MEC dataset (MEC-simulator), PETA, PEdesTrian Attribute:	EDITORS method results in a significantly shorter user-experienced training time, by	The paper did not investigate security mechanisms, model generalization, and

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 7, Issue 3, 2024 (pp. 107-117)
www.abjournals.org

	Reinforcement Transfer Learning in SDN-enabled Edge Nodes	Energy-aware Decision Making	Task arrival rates, computational resource availability, network conditions, and energy consumption profiles	60%–70% compared to existing offloading methods. This efficient training time translates to substantial energy savings in practical implementations and consumption.	thorough scalability studies.
8	A decision-making mechanism for task offloading using learning automata and deep learning in mobile edge networks	Learning automata, DRL-based Asynchronous Advantage Actor-Critic (AAA-C) algorithm, Fuzzy-based A3C algorithm and Auto-Scaling with Long Short-Term Memory (LSTM)	iFogSim simulator: Energy utilized, edge server uplink rate, processing rate (edge server), processing rate (cloud server), giga cycles, data rate and cloud server	The proposed method has improved CPU consumption, execution time, and energy consumption compared to existing equivalent methods such as LAF and LAQ. The superiority of the method is proven on different workload patterns.	The paper neither cover workload prediction using DRL approaches nor multi-user distributed MECC environment.

RESULTS AND DISCUSSION

The average weighted response time incurred by Akhavan et al. (2022) in DECENT increases, while still maintaining a low level (< 500ms). The average weighted response time increment of the other algorithms is much larger than DECENT, which demonstrates that DECENT achieves better offloading decision and resource management in both light and heavy workload scenarios. To achieve the optimization goals of low energy consumption, low latency, and load balancing for edge computing offload policies in an environment where the network state and task demands are dynamically changing (Zhu et al., 2023). The proposed method by Aghasi and Rituraj (2022) in their paper not only made more optimal decisions, but also acted faster in reaching the optimal policy than conventional SARSA algorithms. QoS-SLA-AGA on the other hand executes requests 1.04, 1.23, 1.05, and 9.41 times faster on average compared to the PSO, random offloading, ACC, and AEC approaches respectively. It has the fastest convergence with the shortest total execution time, it also gives a higher total execution time (Communications, 2023).

EDITORS' method results in a significantly shorter user-experienced training time, with a reduction of training time by 60%–70% compared to existing offloading methods because of the TL module in the DRTL framework. This efficient training time translates to substantial energy

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

savings in practical implementations and consumption (Baker et al., 2024). Niu et al. (2023) show that the task-offloading performance of the proposed pipelining task-offloading strategy generated by the delay-aware MADRL algorithm outperforms the traditional MADRL algorithm in random delay Cyber Twin-enabled 6G networks. Furthermore, the proposed pipelining task offloading strategy also performs better in terms of reaction time.

Simulation results demonstrated that the genetic algorithm developed by Chakraborty (2022) effectively reduced task completion times and energy consumption compared to traditional task offloading methods. The use of GAs allowed for a more balanced and efficient distribution of tasks across the available resources in the MEC network. By leveraging genetic algorithms, this research contributes to more sustainable and efficient MEC operations, providing insights into how advanced optimization techniques can enhance the performance of edge computing systems.

CONCLUSION

In IoT devices and multi-access networks, task offloading is a nontrivial decision-making problem that significantly affects the performance and dependability of newer generations of devices and technologies. Reinforcement learning is used in solving such decision-making problems, because it can function in the absence of a model by utilizing data from the environment. The delay that might arise during data acquisition from the environment affects the successful implementation of a good policy. This paper studied different task-offloading algorithms and the techniques used to improve delay, energy or other resource consumption during data computation. Performance of IoT devices will greatly improve with the optimization of edge computing which supports computation of tasks in these devices, specifically by minimizing latency or round trip time in task offloading.

REFERENCES

1. Bello, O., & Zeadally, S. (2022). Internet of Things: Applications, Challenges, and Opportunities. *Future Internet*, *14*(1), 1-18. https://doi.org/10.3390/fi14010001.
2. Jiang, X., & He, Y. (2023). Cloud Computing: Evolution, Applications, and Challenges. *Journal of Cloud Computing: Advances, Systems and Applications*, *12*(1), 1-15. https://doi.org/10.1186/s13677-023-00276-0.
3. Wang, X., Chen, M., Taleb, T., Ksentini, A., & Leung, V. C. M. (2021). Cache in the air: Exploiting content caching and delivery techniques for 5G systems. *IEEE Communications Magazine*, 59(6), 62-68.
4. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. *MIT Press*.
5. Li, J., Xu, H., Zhang, L., & Liu, Y. (2024). Deep reinforcement learning for dynamic task offloading in edge computing. *IEEE Transactions on Network Science and Engineering*, 19(8), 2027.
6. AlShathri, S. I., Hassan, D. S. M., & Chelloug, S. A. (2023). Latency-aware dynamic second offloading service in SDN-based fog architecture. *Computers, Materials & Continua*, *75*(1), 1501-1526. https://doi.org/10.32604/cmc.2023.035602.

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG

7. Aghasi, A., & Rituraj, R. (2022). A Latency-Aware Power-efficient Reinforcement Learning Approach for Task Offloading in Multi-Access Edge Networks. *CANDO-EPE 2022 - Proceedings: IEEE 5th International Conference and Workshop in Obuda on Electrical and Power Engineering*, 181–186. https://doi.org/10.1109/CANDO-EPE57516.2022.10046357.
8. Akhavan, Z., Esmaeili, M., Badnava, B., Yousefi, M., Sun, X., Devetsikiotis, M., & Zarkesh-Ha, P. (2022). Deep Reinforcement Learning for Online Latency Aware Workload Offloading in Mobile Edge Computing. *Proceedings - IEEE Global Communications Conference, GLOBECOM, Mcc*, 2218–2223. https://doi.org/10.1109/GLOBECOM48099.2022.10001678.
9. Baker, T., Al Aghbari, Z., Khedr, A. M., Ahmed, N., & Girija, S. (2024). EDITORS: Energy-aware Dynamic Task Offloading using Deep Reinforcement Transfer Learning in SDN-enabled Edge Nodes. *Internet of Things (Netherlands)*, *25*(January). https://doi.org/10.1016/j.iot.2024.101118.
10. Chakraborty, S. (2022). *Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing*. *34*(4), 1552–1568.
11. Dayong, W., Bin, K., Bakar, A., Isyaku, B., Abdalla, T., Eisa, E., & Abdelmaboud, A. (2024). *Heliyon Review article A comprehensive review on Internet of things task offloading in multi-access edge computing*. *10*(February).
12. Dikonimaki, C. (2023). *Edge Compute Offloading Strategies using Heuristic and Reinforcement Learning Techniques. Strategies using Heuristic*.
13. Gao, L., Li, W., & Ma, H. (2024). *High-Confidence Computing Data cube-based storage optimization for resource-constrained edge computing*. *February*, 1–16.
14. Hua, H., Li, Y., Wang, T., Dong, N., Li, W. E. I., & Cao, J. (2023). *Edge Computing with Artificial Intelligence : A Machine*. *55*(9). https://doi.org/10.1145/3555802.
15. Huang, L., Feng, X., Zhang, C., Qian, L., & Wu, Y. (2019). Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing. *Digital Communications and Networks*, *5*(1), 10–17. https://doi.org/10.1016/j.dcan.2018.10.003.
16. Huang, Y., Ma, X., Fan, X., Liu, J., & Gong, W. (2017). *When Deep Learning Meets Edge Computing*.
17. Nguyen, T., Nguyen, H., & Nguyen, T. (2024). *Journal of Network and Computer Applications Exploring the integration of edge computing and blockchain IoT : Principles, architectures, security, and applications*. *226*(March).
18. Wang, F., & Ma, X. (2020). Deep Learning for Edge Computing Applications : A State-of-the-Art Survey. *IEEE Access*, *8*, 58322–58336. https://doi.org/10.1109/ACCESS.2020.2982411.
19. Zhu, X. (2023). *Deep reinforcement learning-based edge computing offloading algorithm for software-defined IoT*. *235*(November), 1–7.

Article DOI: 10.52589/BJCNIT-UGHJH8QG
DOI URL: https://doi.org/10.52589/BJCNIT-UGHJH8QG