Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 11 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i11.49051

PAPER

Secured Computation Offloading in Multi-Access Mobile Edge Computing Networks through Deep Reinforcement Learning

Rijal Abdullah[1], Noorulsadiqin Azbiya Yaacob[2](✉), Anas A. Salameh[3], Nur Amalina Mohamad Zaki[4], Nur Fadhilah Bahardin[5]

[1]Faculty of Engineering, Universitas Negeri Padang, Padang, Indonesia

[2]School of Technology Management and Logistics, Universiti Utara Malaysia, Kedah, Malaysia

[3]Department of Management Information Systems, College of Business Administration, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

[4]Faculty of Business, Economics and Social Development, Universiti Malaysia Terengganu, Terengganu, Malaysia

[5]Department of Built Environment and Technology, College of Built Environment, UiTM Perak Branch, Perak, Malaysia

sadiqin@uum.edu.my

ABSTRACT

Mobile edge computing (MEC) has emerged as a pivotal technology to address the computational demands of resource-constrained mobile devices by offloading tasks to nearby edge servers. However, ensuring the security and efficiency of computation offloading in multi-access MEC networks remains a critical challenge. This paper proposes a novel approach that leverages deep reinforcement learning (DRL) for secure computation offloading in multi-access MEC networks. The proposed framework utilizes DRL agents to dynamically make offloading decisions based on the current network conditions, resource availability, and security requirements. The agents learn optimal offloading policies through interactions with the environment, aiming to maximize task completion efficiency while minimizing security risks. To enhance security, the framework integrates encryption techniques and access control mechanisms to protect sensitive data during offloading. The proposed approach undergoes comprehensive simulations to assess its performance in terms of security, efficiency, and scalability. The results demonstrate that the DRL-based approach effectively balances the trade-offs between security and efficiency, achieving robust and adaptive computation offloading in multi-access MEC networks. This study contributes to advancing the state-of-the-art in secure and efficient mobile edge computing systems, fostering the development of intelligent and resilient MEC solutions for future mobile networks.

KEYWORDS

mobile edge computing (MEC), security, multi-access networks, deep reinforcement learning (DRL), computation offloading, resource allocation, task efficiency

1 INTRODUCTION

The proliferation of mobile devices and the exponential growth of data traffic have catalyzed a paradigm shift in wireless communication networks [1]. Traditional

Abdullah, R., Yaacob, N.A., Salameh, A.A., Zaki, N.A.M., Bahardin, N.F. (2024). Secured Computation Offloading in Multi-Access Mobile Edge Computing Networks through Deep Reinforcement Learning. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(11), pp. 80–91. https://doi.org/10.3991/ijim.v18i11.49051

Article submitted 2024-02-05. Revision uploaded 2024-03-14. Final acceptance 2024-03-14.

centralized computing architectures, such as cloud computing, are struggling to meet the increasing demands for low-latency, high-throughput applications. To address these obstacles, mobile edge computing (MEC) has emerged as a promising paradigm to enhance the capabilities of wireless networks by bringing computation closer to the network edge. By deploying computational resources, storage, and networking infrastructure at the edge of the network, MEC aims to alleviate the burden on centralized cloud servers and reduce latency for time-sensitive applications [2], [3].

At the core of MEC lies the concept of computation offloading, where computationally intensive tasks are transferred from resource-constrained mobile devices to nearby edge servers for processing. Offloading computations to edge servers can significantly enhance the performance of mobile applications by leveraging the proximity of computational resources and reducing the communication latency between devices and servers [4]. However, the seamless integration of computation offloading into MEC networks presents a myriad of challenges, particularly concerning security, privacy, and resource management [5], [7].

One of the primary concerns in computation offloading is the security of sensitive data during transmission and processing. Mobile devices often store a wealth of personal and confidential information, including financial transactions and healthcare records, which makes them prime targets for malicious attacks. Offloading computations to remote edge servers introduces additional security risks as data traverses potentially untrusted network channels and resides on external servers. Ensuring the confidentiality, integrity, and authenticity of data in transit and at rest is paramount to preventing unauthorized access and data breaches [6], [8].

Furthermore, the dynamic and heterogeneous nature of MEC networks exacerbates the security challenges associated with computation offloading. Multi-access MEC networks (M-MEC) integrate diverse access technologies, including Wi-Fi, cellular and other wireless standards, to provide ubiquitous connectivity and seamless handover between network domains. The heterogeneity of access technologies introduces complexities in security management and enforcement. Different access networks may have varying levels of trustworthiness and security capabilities. An MEC network is organized into three layers: the user layer, edge layer, and cloud layer, as illustrated in Figure 1.

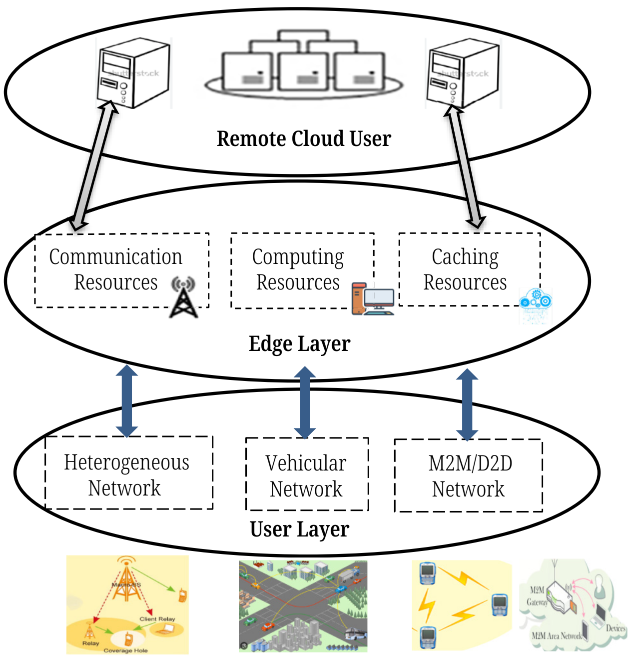

Fig. 1. MEC network architecture

User layer: The user layer includes a variety of IoT devices, such as smartphones, laptops, vehicles, and sensors, each requiring computational capabilities. These computations are wirelessly transmitted to the edge layer and divided into separate networks. The first network, a heterogeneous network, caters to devices that require high data rates. It features numerous small macro-base stations strategically located in dense areas to enhance connectivity and minimize mobile device power consumption. These base stations are equipped with robust computing resources for task offloading. The second network, a vehicular network, includes transportation units, pedestrians, and roadside units. Roadside units, equipped with computing capabilities, facilitate communication among transportation units, pedestrians, and nearby infrastructure for traffic safety and regulation enforcement. This network supports smart applications such as in-car media streaming and parking assistance. Finally, the device-to-device (D2D) network enables peer-to-peer communication among IoT devices via wireless links. This network decentralizes computing capabilities, allowing devices to offload tasks to other devices as well as edge servers [9].

Edge layer: The edge layer consists of distributed servers with advanced computing power strategically deployed at locations such as subway stops, highways, and airports to reduce latency. These servers aim to efficiently handle time-sensitive and computing-intensive tasks from the user layer, requiring enhanced communication resources for efficient task resolution.

Cloud layer: The cloud layer connects multiple edge servers, enabling data mining for neural network training and efficient resource allocation. It stores extensive network metadata, which reduces the load on edge servers. This layer enhances the management and security of edge servers, ensuring optimal performance and the protection of network assets [10].

In addition to security concerns, privacy preservation emerges as a critical consideration in computation offloading. Mobile users are becoming more aware of their privacy rights and are demanding guarantees that their personal data is handled transparently and securely. Offloading sensitive computations to external servers raises privacy implications, as third-party entities may have access to sensitive information without users' explicit consent. Balancing the benefits of computation offloading with the privacy concerns of mobile users requires the development of privacy-preserving techniques that safeguard sensitive data while enabling efficient offloading.

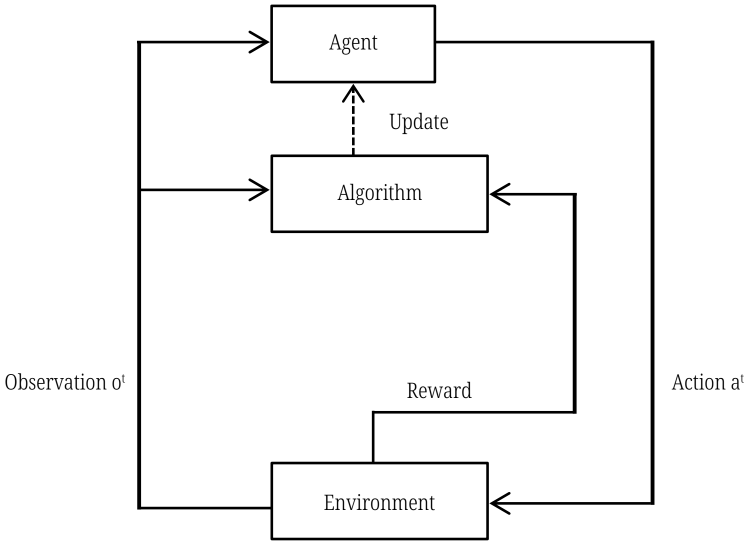

Fig. 2. Interaction model of RL

Within reinforcement learning (RL), the agent interacts with the environment through trial and error, aiming to optimize rewards while minimizing penalties. The agent learns from its experiences by storing them in a Q-table, which maps states to actions and their associated rewards. When the agent observes a state S_t at time step t, it takes action A_t to transition to the next state S_{t+1} and receives a reward R_{t+1}. This process enables the agent to gradually improve its decision-making abilities based on past interactions, as illustrated in Figure 2.

2 LITERATURE REVIEW

[11] Proposed an innovative design for a blockchain-based multi-UAV-enabled MEC system. This system focuses on ensuring secure computational offloading and resource allocation within IoT networks by utilizing a deep reinforcement learning (DRL) approach in an A2G network infrastructure. In this setup, UAVs serve as aerial base stations, providing support to overloaded base stations and replacing those that are damaged during natural or human-made disasters.

A novel approach for model-free DRL was suggested by [12], integrating an asynchronous advantage actor-critic (A3C) algorithm. This method aims to optimize offloading decisions efficiently. Through extensive numerical experimentation, it was demonstrated that the A3C algorithm significantly enhances the convergence rate of the system while simultaneously mitigating the overall energy consumption of GME (generic mobile edge).

[13] Introduced a novel approach called DRCOM (deep reinforcement learning for computation offloading in a UAV-assisted multi-access edge computing network), leveraging UAVs as aerial base stations. By employing deep reinforcement learning, the method addresses the challenge of determining the computation offloading policy, enabling resource allocation aimed at maximizing computing performance.

[15] The DROO algorithm was proposed, utilizing deep reinforcement learning for online offloading in wireless-powered MEC networks. The aim is to enhance the rate of weighted sum computation through binary computation offloading. To expedite algorithm convergence, they developed an order-preserving quantization technique and an adaptive parameter setting method. Simulation findings indicate that DROO achieves performance close to optimal compared to existing benchmark methods while significantly decreasing CPU execution latency by over tenfold. This progress enables the possibility of real-time optimization for wireless-powered MEC networks, even in environments affected by fading.

[16] The study introduced a deep reinforcement learning-based offloading scheme for XR devices (DRLXR) in a MEC-enabled network framework. The offloading challenge in XR devices is addressed through DRL formulation. Utilizing data such as radio signal strength, energy usage, and application status monitored by XR devices, an actor-critic approach is employed for training and decision-making regarding task offloading. Simulation outcomes demonstrate the superior performance of DRLXR compared to alternative solutions, particularly in terms of average energy consumption and overall completion time.

3 METHODOLOGY

To realize the proposed approach of leveraging DRL for secure computation offloading in multi-access MEC networks, a systematic methodology is employed.

The methodology comprises the design, implementation, and evaluation stages, incorporating key elements to ensure the robustness, adaptability, and security of the computational offloading framework. Figure 3 illustrates the proposed DQN framework for deep reinforcement learning.

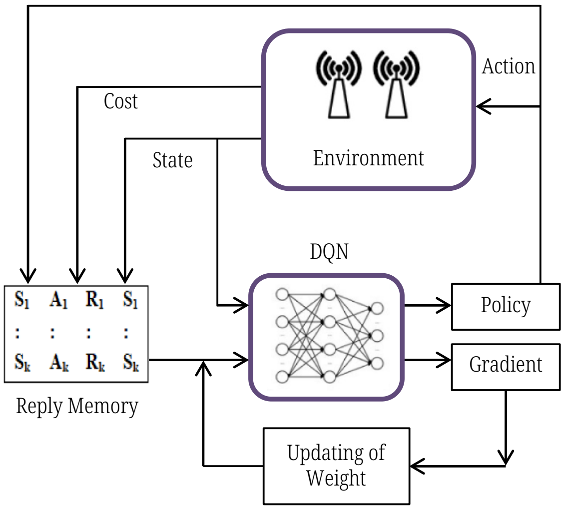

Fig. 3. Proposed architecture of DQN-based system

4 PROBLEM FORMULATION

Within the MEC framework, Figure 3 illustrates a setup where multiple users are connected to a base station with a high-performance server. These servers, strategically positioned at the network's edge, offer cloud-like services in close proximity to end-users. Consequently, users choose to delegate their computationally demanding tasks to the nearby edge server they are connected to rather than relying solely on a distant cloud infrastructure. This approach significantly reduces latency, enabling a diverse range of applications and services to operate with real-time or near-real-time performance. By decentralizing computation to the network edge, MEC enhances scalability, flexibility, and accessibility. This architecture enables users to efficiently utilize edge computing resources, allowing seamless access to a wide range of services in sectors such as healthcare, smart cities, industrial automation, and multimedia streaming.

This paper assumed a group of devices, denoted as

$$N = \{N_1, N_2, \ldots, N_k\} \tag{1}$$

In the problem formulation presented, a group of k mobile users with single antennas establish communication with a large-scale MEC server via a base station featuring N antennas (where N is significantly greater than k). Furthermore, when multiple devices engage in offloading simultaneously, the available bandwidth, denoted as B, will be distributed equally among them.

Each device's task is represented as

$$DT_i = (W_i, P_i) \qquad i \in N \tag{2}$$

Where, W_i signifies the magnitude of the computation that the device seeks to address, while P_i represents the total number of CPU cycles needed to accomplish the task. There exists a positive correlation between both P_i and W_i, with P_i remaining constant throughout the computation process. Ultimately, task parameters are scheduled through task profiles derived from an application, and these parameters may vary among different applications.

When a device chooses to execute a task locally, it relies on the local computing model. The delay experienced during local execution is identified as

$$D_i^1 = \frac{P_I}{C_i} \tag{3}$$

It consists of the aggregate CPU cycles, P_i, and the computational capability of the CPU, represented as C_i. Essentially, D_i indicates the duration required by the device to complete R_i, a duration that may vary among devices depending on the computing capacity of their local CPUs.

The calculation of the energy consumed to complete R_i is determined by

$$E_i^1 = y_i P_i \tag{4}$$

The energy consumption formula comprises the energy usage per CPU cycle, labeled as y_i, along with the cumulative CPU cycle count required.

The overall cost of local computing can be calculated accordingly,

$$C_i^1 = W_i^t D_i^1 + W_i^e E_i^1 \tag{5}$$

It is derived from Equations (4) and (5). Here, W_i^t and W_i^e stand for the weights of time and energy.

5 OFFLOADING MODEL

The offloading computation model is applied when a device chooses to transfer a task via wireless communication to the MEC server. Initially, the upload rate for the devices is computed based on,

$$R_i = \frac{B}{K} log\left(1 + \frac{T_p G_i}{\frac{B}{K} N_0}\right) \tag{6}$$

The initial parameter here is the bandwidth, denoted as B, distributed among all K devices in the system that choose to offload. T_p represents the transmission power, N_0 stands for the variance of complex white Gaussian channel noise, and G_i signifies the channel gain for the wireless channel. Once the upload rate is determined, each device initiates the process of uploading input parameters to the base station before transmitting the computation task to the MEC server. The calculation of transmission delay can be depicted as follows:

$$T_{i,t}^0 = \frac{W_i}{R_i} \tag{7}$$

$T_{i,t}^0$ indicates the duration required for uploading the computation task via wireless communication to the MEC server. After determining the delay from the device to the server, the processing delay of the MEC server can be computed using the equation provided below.

$$T_{i,p}^{0} = \frac{P_i}{C_i} \tag{8}$$

Here, $T_{i,p}^{0}$ signifies the duration required by the edge server to process and finalize the task transmitted by the device [14]. It refers to the time taken for the server to execute computational operations and deliver the results back to the device. Additionally, the parameter C_i denotes the allocation of resources from the MEC server specifically designated to complete the task on behalf of the offloading device. This resource allocation ensures that the task is efficiently executed on the server end, optimizing processing time and overall performance. The algorithm for the proposed DQN approach is presented below (see Algorithm 1).

Algorithm 1: DQN Algorithm

```
Initialize replay memory D with capacity N
Initialize Q-network with random weights θ
Initialize target Q-network with weights θ_target = θ
Initialize offloading environment
for episode = 1, M do
  Initialize state s
    for t = 1, T_max do
    Choose action a from state s using ε-greedy policy
    'Execute action a and observe reward r and next state s'
    'Store transition (s, a, r, s') in replay memory D'
    'Sample random mini-batch of transitions (sj, aj, rj, s'j) from D'
    Compute target Q-values:
      if s' is the terminal state then
        target = rj
      else
        target = rj + γ * max(Q_target(s'_j, a', θ_target))
    Update Q-network parameters θ by minimizing the loss:
      loss = 1/N * Σ(target – Q(sj, aj, θ))²
      θ = θ – α * ∇_θ(loss)
    For every C steps, update target Q-network: θ_target = θ
    if s' is the terminal state then
      Break
    else
      s = s'
  Every episode, evaluate performance and monitor convergence
end for
```

M denotes the total number of episodes. T_max represents the maximum number of steps per episode. N represents the capacity of the replay memory. α represents the learning rate. γ is the discount factor. C represents the frequency of updating the target Q-network. Q(s, a, θ) represents the Q-value function of the Q-network with parameters θ. Q_target(s', a', θ_target) represents the Q-value function of the target Q-network with parameters θ_target. The ε-greedy policy is used to balance exploration and exploitation.

The offloading environment defines the state space, action space, and reward structure specific to the computation offloading problem in multi-access MEC networks.

6 RESULTS AND DISCUSSION

This section presents the simulation results, including a comparison of performance with other baseline methods. The comparative analysis of the DRL-DQN, DRL-A3C, and DRCOM approaches reveals compelling advantages of the DRL-DQN framework across key performance metrics, including security, efficiency, and scalability. The results, as depicted in the graphs below, illustrate the superiority of the DRL-DQN framework over DRL-A3C and DRCOM in various aspects.

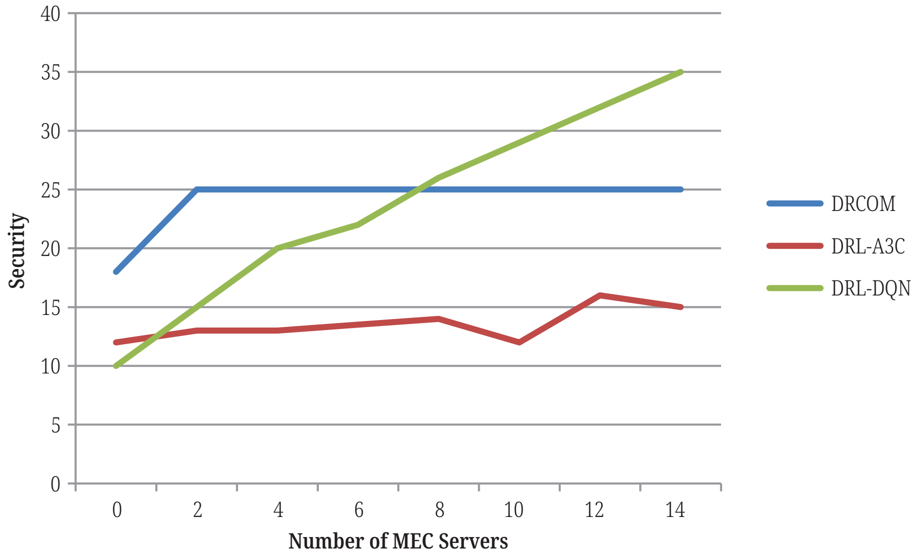

Fig. 4. Comparison of security with number of MEC servers

The security graph illustrates that the DRL-DQN framework consistently outperforms both DRL-A3C and DRCOM in ensuring data confidentiality and integrity during computation offloading processes, as depicted in Figure 4. Through advanced encryption techniques and access control mechanisms, the DRL-DQN framework achieves higher security levels, mitigating potential cyber threats and unauthorized access attempts.

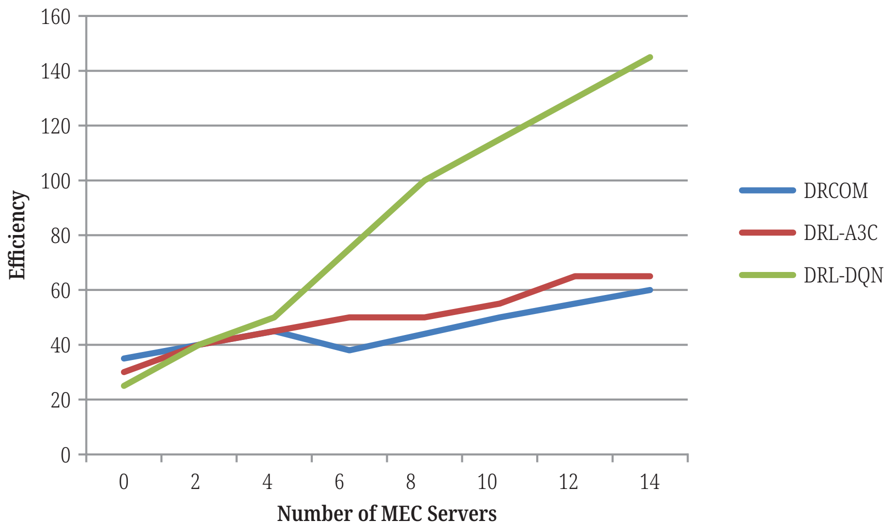

Fig. 5. Comparison of efficiency with number of MEC servers

In terms of efficiency, the graph illustrates that the DRL-DQN framework demonstrates superior task completion rates and reduced computational overhead compared to DRL-A3C and DRCOM, as evaluated in Figure 5. Through dynamic optimization of resource allocation and offloading decisions, the DRL-DQN framework minimizes processing delays and enhances overall system efficiency, resulting in faster task execution and an improved user experience.

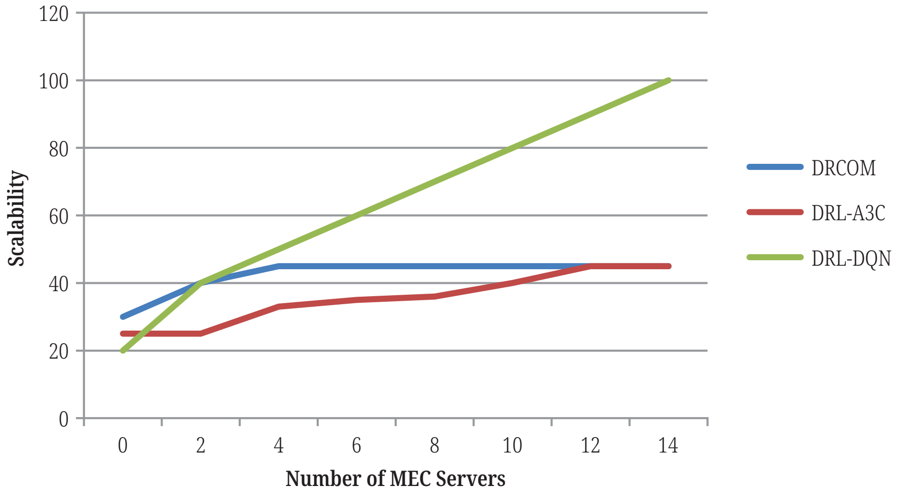

Fig. 6. Comparison of scalability with number of MEC servers

The scalability graph illustrates the scalability of the DRL-DQN framework across different numbers of MEC servers, as shown in Figure 6. Unlike DRL-A3C and DRCOM, which may experience performance degradation with increasing server deployments, the DRL-DQN framework maintains consistent performance levels and adapts seamlessly to changes in network scale. This scalability advantage ensures that the framework remains effective and efficient in large-scale MEC deployments, accommodating growing numbers of devices and tasks without compromising security or efficiency.

The comparison graphs demonstrate that the DRL-DQN framework offers superior security, efficiency, and scalability compared to the DRL-A3C and DRCOM approaches. These findings underscore the effectiveness of the DRL-DQN framework in addressing the challenges of computation offloading in multi-access MEC networks, paving the way for secure, efficient, and scalable mobile edge computing solutions.

7 CONCLUSION

This study introduces a novel framework for secure computation offloading in multi-access MEC networks using DRL. The framework utilizes DRL agents to dynamically make offloading decisions based on real-time network conditions, resource availability, and security requirements. By learning optimal offloading policies through interactions with the environment, the agents aim to maximize task completion efficiency while minimizing security risks. To enhance security, encryption techniques and access control mechanisms are integrated into the framework to protect sensitive data during offloading. Comprehensive simulations evaluate the framework's performance in terms of security, efficiency, and scalability. Results indicate that the DRL-DQN-based approach effectively balances trade-offs between security and efficiency, achieving robust and adaptive computation offloading in multi-access MEC networks. Compared to existing methods, the framework demonstrates superior performance in security, efficiency, and scalability. This study contributes to advancing the state-of-the-art in secure and efficient mobile edge computing systems, facilitating the development of intelligent and resilient MEC solutions for future mobile networks.

8 FUNDING STATEMENT

This study is supported by funding from Prince Sattam bin Abdulaziz University, project number (PSAU/2023/R/1445).

9 REFERENCES

[1] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, and N. Georgalas, "Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 64–69, 2019. https://doi.org/10.1109/MCOM.2019.1800971

[2] J. Park and K. Chung, "Distributed DRL-based computation offloading scheme for improving QoE in edge computing environments," *Sensors*, vol. 23, no. 8, p. 4166, 2023. https://doi.org/10.3390/s23084166

[3] K. Wang, X. Wang, and X. Liu, "A high reliable computing offloading strategy using deep reinforcement learning for IoVs in edge computing," *Journal of Grid Computing*, vol. 19, 2021. https://doi.org/10.1007/s10723-021-09542-6

[4] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning based resource allocation for UAV networks," *arXiv preprint arXiv:1810.10408*, 2018.

[5] Y. He, Z. Zhang, and Y. Zhang, "A big data deep reinforcement learning approach to next generation green wireless networks," in *IEEE Global Communications Conference (GLOBECOM 2017)*, 2017, pp. 1–6. https://doi.org/10.1109/GLOCOM.2017.8254717

[6] P. Zhang, C. Wang, C. Jiang, and A. Benslimane, "UAV-assisted multi-access edge computing: Technologies and challenges," *IEEE Internet Things Magzine*, vol. 4, no. 4, pp. 12–17, 2021. https://doi.org/10.1109/IOTM.001.2100092

[7] M. El Ghmary, S. Ziani, T. Chanyour, and Y. Hmimz, "An energy and latency trade-off for resources allocation in a MEC system," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 20, pp. 133–147, 2023. https://doi.org/10.3991/ijim.v17i20.40999

[8] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 14–19, 2018. https://doi.org/10.1109/MCOM.2018.1701069

[9] A. Robles-Enciso and A. F. Skarmeta, "A multi-layer guided reinforcement learning-based tasks offloading in edge computing," *Computer Network*s, vol. 220, p. 109476, 2023. https://doi.org/10.1016/j.comnet.2022.109476

[10] F. Alsokhiry, A. Annuk, M. A. Mohamed, and M. Marinho, "An innovative cloud-fog-based smart grid scheme for efficient resource utilization," *Sensor*s, vol. 23, no. 4, p. 1752, 2023. https://doi.org/10.3390/s23041752

[11] A. Mohammed, H. Nahom, A. Tewodros, Y. Habtamu, and G. Hayelom, "Deep reinforcement learning for computation offloading and resource allocation in blockchain-based multi-UAV-enabled mobile edge computing," in *17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2020, pp. 295–299. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317445

[12] B. Li, S. Yu, J. Su, J. Ou, and D. Fan, "Computation offloading in multi-UAV-enhanced mobile edge networks: A deep reinforcement learning approach," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–11, 2022. https://doi.org/10.1155/2022/6216372

[13] N. Lin *et al.*, "Deep reinforcement learning-based computation offloading for servicing dynamic demand in multi-UAV-assisted IoT network," *IEEE Internet of Things Journal*, 2024. https://doi.org/10.1109/JIOT.2024.3356725

[14] S. Maftah, M. El Ghmary, H. El Bouabidi, M. Amnai, and A. Ouacha, "Optimal task processing and energy consumption using intelligent offloading in mobile edge computing," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 16, no. 20, pp. 130–142, 2022. https://doi.org/10.3991/ijim.v16i20.34373

[15] L. Huang, S. Bi, and Y. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, 2020. https://doi.org/10.1109/TMC.2019.2928811

[16] B. Trinh and G. M. Muntean, "A deep reinforcement learning-based offloading scheme for multi-access edge computing-supported eXtended reality systems," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1254–1264, 2023. https://doi.org/10.1109/TVT.2022.3207692

10 AUTHORS

Dr. Rijal Abdullah, MT. Solok works as a Lecturer at the Faculty of Engineering, Padang State University. The author is active in writing articles in the fields of engineering and education (E-mail: rijal_a@ft.unp.ac.id).

Dr. Noorulsadiqin Azbiya Yaacob works as an Associate Professor at Universiti Utara Malaysia (UUM) in the School of Technology Management and Logistics. She graduated from UUM in 2000 with a Bachelor of Information Technology with Honors and in 2004 with a Master of Science in Information Technology through research, focusing on network security. She continued her education and graduated from Universiti Teknologi Malaysia (UTM) in 2011 with a Ph.D. in Technology Management. Her area of expertise is innovation management, with a focus on utilizing information and communication technology (ICT) to address contemporary, multidisciplinary issues sustainably. The design and development of mobile apps are the main focus of her current research. She works to promote innovation in mobile app development, contributing to a more sustainable and user-friendly environment (E-mail: sadiqin@uum.edu.my).

Dr. Anas A. Salameh is an Associate Professor in the Department of Management Information Systems at the College of Business Administration, Prince Sattam Bin Abdulaziz University. He has held this position since 2016. Additionally, he serves as the current deputy director of the students' activities committee and is a member of the exams scheduling committee at PSAU. His major research interests focus on areas such as e-commerce (m-commerce), e-business, e-marketing, technology acceptance/adoption, e-learning, e-CRM, and service quality. He has evaluated service quality in various areas related to e-services aspects (E-mail: a.salameh@psau.edu.sa).

Nur Amalina Mohamad Zaki is a Senior Lecturer at the Faculty of Business, Economics, and Social Development (FBESD) at Universiti Malaysia Terengganu (UMT). Amalina received her bachelor's degree in Business Administration with triple majors in Management, International Business, and Business Information Systems from Indiana University, Bloomington, USA. She then pursued her Master of Business Administration with a specialization in Marketing from Western Michigan University in Michigan, USA. Amalina holds a PhD in Marketing from Griffith

University, Australia in 2017 with a doctorate study titled "An Exploratory Study Social Media Role in Business-to-business Relationship Marketing in Malaysia". She also went for a post-doctoral degree at University of Sydney, Australia in 2018. Her research interests are in the field of relationship marketing. Currently, she holds several domestic and international research grants (E-mail: amalina@umt.edu.my).

Nur Fadhilah Bahardin is a Senior Lecturer of Programme in the Building Surveying, Department of Built Environment Studies and Technology, College of Built Environment, UiTM Perak Branch. She is also a graduate member of the Royal Institution of Surveyors Malaysia (RISM). She received her M.Sc. in Facilities Management in 2010 from UiTM. Nur Fadhilah teaches several courses on Building Surveying, with a focus on Measurement and Estimating, Building Surveying, and Maintenance. Her research interests and specialization are in Digitalization in Built Heritage, Building Performance Evaluation, Building Pathology, and Forensics (E-mail: nurfa644@uitm.edu.my).