Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 17 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i17.50729

PAPER

Deep Reinforcement Learning Based Secure Transmission for UAV-Assisted Mobile Edge Computing

N. Vijayalakshmi[1](✉), Sagar Gulati[2], B. Ben Sujin[3], B. Madhav Rao[4], K. Kiran Kumar[5]

[1]Department of Computer Science and Applications, SRMIST, Ramapuram Campus, Chennai, Tamil Nadu, India

[2]School of Computer Science and IT, JAIN Deemed to be University, Bengaluru, Karnataka, India

[3]Computer Engineering Department, University of Technology and Applied Sciences, Nizwa, Sultanate of Oman

[4]Department of CSE, SIR C R Reddy College of Engineering, Eluru, Andhra Pradesh, India

[5]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India

vijayaln@srmist.edu.in

ABSTRACT

The increasing computational demand for real-time mobile applications has led to the development of mobile edge computing (MEC), with support from unmanned aerial vehicles (UAVs), as a promising paradigm for constructing high-throughput line-of-sight links for ground users and pushing computational resources to network edges. Users can reduce processing latency and the load on their local computers by delegating tasks to the UAV in its role as an edge server. The coverage capacity of a single UAV is, however, very limited. Moreover, it will be easy to intercept the data that is transferred to the unmanned aerial vehicle. Thus, for UAV-assisted mobile edge computing, we proposed a transmission technique based on multi-agent deep reinforcement learning in this study. The recommended approach to maximize UAV deployment first applies the particle swarm optimization algorithm. Then, deep reinforcement learning is utilized to optimize the secure offloading to maximize the system utility and minimize the quantity of information eavesdropping, taking into consideration different user task types with diverse preferences for processing time and residual energy of computing equipment. The results of the simulation demonstrate that, in comparison to the single-agent strategy and the benchmark, the multi-agent approach can optimize offloading more successfully and produce higher system utility.

KEYWORDS

mobile edge computing (MEC), unmanned aerial vehicle (UAV) communication, secure transmission, deep reinforcement learning (DRL)

1 INTRODUCTION

1.1 Overview of mobile edge computing

One prospective technology to address these needs is mobile edge computing (MEC), which can give immediate data by putting computing power close to consumers on the move at the network edge. By leveraging their proximity to

Vijayalakshmi, N., Gulati, S., Sujin, B.B., Rao, B.M., Kumar, K.K. (2024). Deep Reinforcement Learning Based Secure Transmission for UAV-Assisted Mobile Edge Computing. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(17), pp. 154–169. https://doi.org/10.3991/ijim.v18i17.50729

Article submitted 2024-05-13. Revision uploaded 2024-06-19. Final acceptance 2024-06-20.

mobile consumers, wireless carriers can enhance the quality of experience (QoE) and offer context-aware offerings in conjunction with RAN information to minimize latency in the network. In addition, peripheral servers can collaborate on service coordination with the main cloud servers. It is anticipated that MEC's decentralization and cooperation will dramatically alter applications and systems. One example of providing cloud service capabilities near the end-user, at the edge of mobile connections, is MEC. The MEC is distinguished by low-end latency, local storage and processing resources, networking perception, and improved quality of service offered by mobile transporters, based on the European Communications Standards Institute. MEC necessitates the smooth integration of resource management and design elements between mobile phone carriers and service suppliers. Several motivations for market change, such as the requirement for mobile phone companies to shorten the introduction period of new, compute-intensive services to boost their earnings, are also driving the development of MEC. As such, the involvement of several stakeholders is necessary to guarantee an effective MEC implementation.

The following differentiates mobile edge computing:

1. **Premises:** MEC is capable of being set up and operated in a completely isolated environment, with no access to any network resources other than those on the local workstation.
2. **Adjacency:** Usually, MEC machines are positioned close to mobile consumers. The implementation of near separation would facilitate the collection, storage, and processing of actual time cellular user data by mobile carriers for many applications, including large-scale analytics and location-aware applications.
3. **Minimal latency:** MEC networks might avoid traffic congestion on front-haul and back-haul network channels by lowering dispersion and latency for communication. This will thereby improve the material and service response time and allow MEC to be a significant enabler for applications for 5G that are latency-sensitive.

The MEC architecture, comprising its processing and communication resources, is covered in this section from the standpoint of dynamic management of resources.

1.2 Mobile edge computing architecture

Mobile customer services are provided by MEC in terms of computation and storage capabilities. It is anticipated that MEC resources will be made accessible through mobile phone networks that are situated near consumers. In particular, MEC resources are relevant to radio networks of access that link consumer devices to the MNO core system, base stations that are located both inside and outside, and connection points. Physical limitations, requirements for performance, and network operators' preferences all influence how MEC assets are precisely deployed. [1] Taking into account the various MEC deployment choices, Figure 1 groups MEC elements according to their roles and how they work together to accomplish the objectives of the MEC network. The first group comprises Internet of Things (IoT) gadgets, mobile phone users, and connected automobiles. Computing components used to manage applications used by consumers fall into the second category. Lastly, the third class

of MEC resources communication components connects the consumer and computer components.

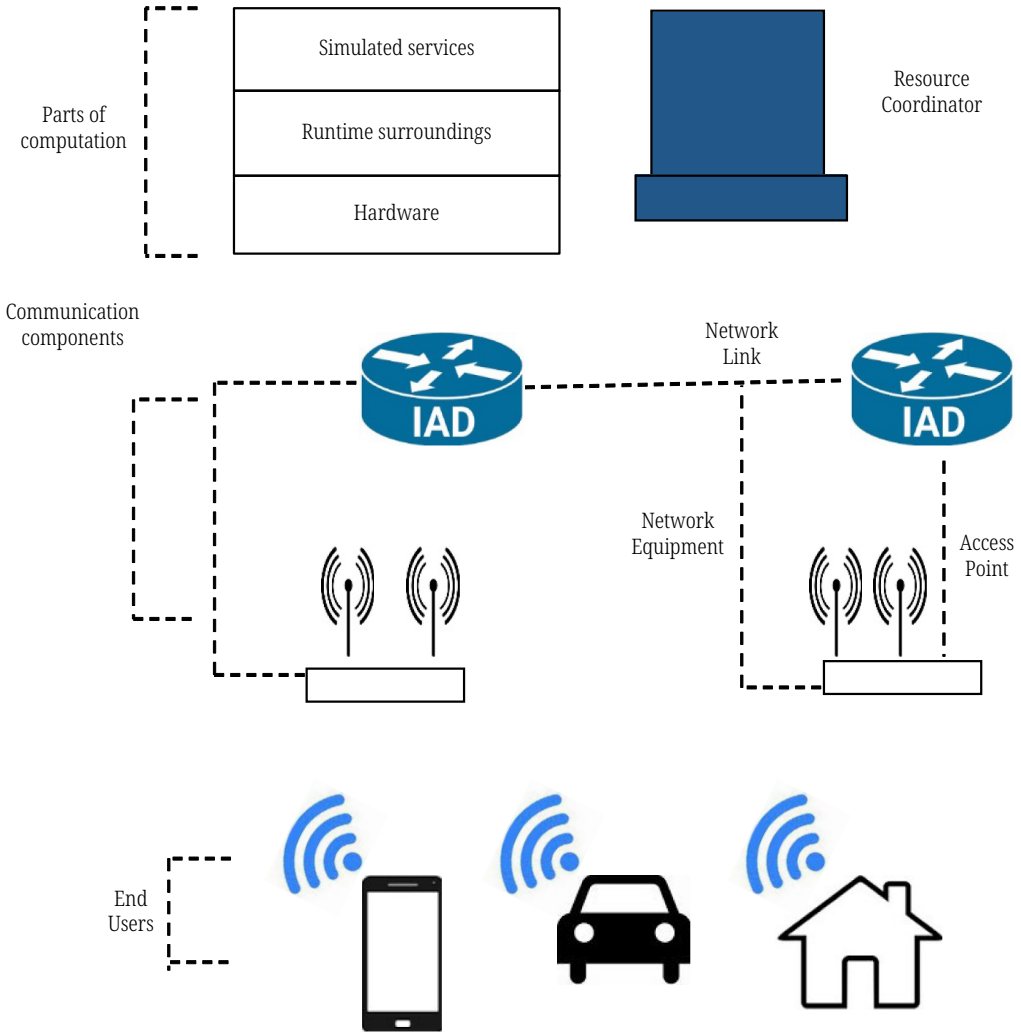

Fig. 1. Mobile edge computing components

1.3 Decentralized mobile edge computing's synchronicity

Massive amounts of data can be processed in real-time, and numerous users can be served by the traditional centralized cloud servers, which have enormous computational and connectivity capacity. Nonetheless, decentralized MEC infrastructure is very sought after since it satisfies QoS customer demands and lowers latencies brought on by delays in transmission and traffic jams. Implementing MEC hierarchically, that is, using the HetNet architecture with three primary layers; mobile users, computing at the edges, and cloud computing is beneficial. To help HetNets disperse compute demands and diversity radio transmissions, the MEC supplier also provides computational power to the smaller eNBs. Concerns regarding delivering latency-critical applications are raised by the observation that decentralized MEC might not have the processing power to handle all user queries.

1.4 Mobile edge computing through UAV-based

The low processing power and short lifespans of IoT devices make it difficult for them to run real-time apps. Fortunately, MEC has been a viable approach to addressing this problem in recent times. By enabling cloud computing features, mobile users can shift their calculation chores to the network's edge with the installation of a MEC server.

It accomplishes two crucial goals.

1. Decrease in application latency when utilizing a distant device with high processing power.
2. Due to the application running remotely on a device, battery life has improved.

Mobile devices with limited resources can transfer their computationally demanding activities to a flying UAV at the edge of the network. This allows the UAV to have high computing power and a variable connection, which saves power and lessens traffic on the centralized cloud servers. [2] In light of this, the MEC server-equipped UAV presents several encouraging benefits over the traditional ground mobile network with fixed base stations. Figure 2 shows an operational model of the MEC installed on a UAV. Each mobile device must choose between edge computing and local computing in this situation. In the case of the former, the embedded processors in mobile devices allow them to carry out their tasks locally, using a lot of energy and local computing resources. About the latter, mobile devices are permitted to transfer their computationally demanding duties directly to the MEC server, which is shared in UAVs; this server will then carry out the computing duties on behalf of the handheld devices. In actuality, every mobile device is linked to a nearby UAV node that has sufficient computational and battery life at the moment. A substantial corpus of study on UAV-based MEC is displayed in Table 1. Using high-performance cloudlets on the deployed UAVs allowed computation and traffic load to be spread from the main cloud to the network edge, reducing both the amount of traffic and the computing load on the cloud servers.

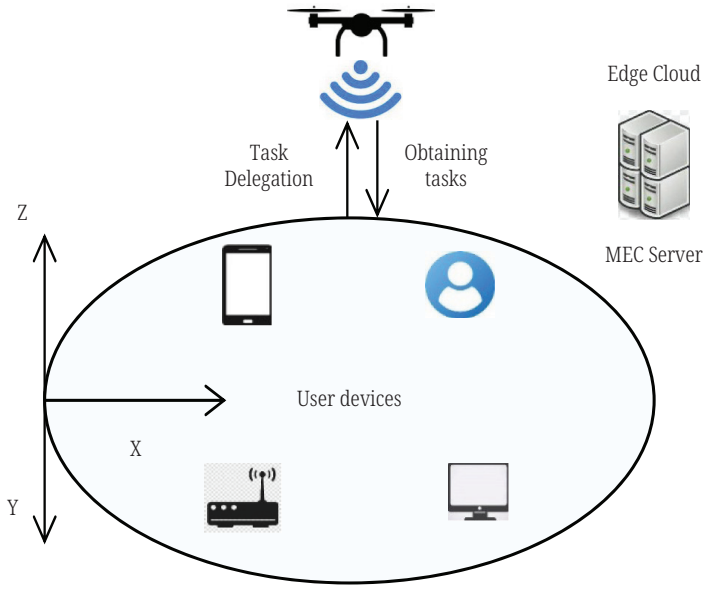

Fig. 2. A diagram showing the MEC system mounted on a UAV

Table 1. A compilation of the implementation of UAV-based MEC

Purpose	Mobility	BSs Types	Number of UAVs
Computation offloading	Mobile	UAV Only	Single UAV
Computation offloading	Mobile	UAV Only	Single UAV
Computation offloading	Mobile	UAV Only	Single UAV
Computation offloading	Mobile	Hybrid UAV	Multiple UAV
Computation offloading	Mobile	UAV Only	Single UAV

1.5 Objectives of optimizing UAV

To take advantage of the benefits of collaboration between UAVs and ECs, we propose a unique UAV-enabled MEC system in which IoT devices use UAVs, which are equipped with computing resources, to offload computing chores to ECs outside of their interaction range. In these parts, a collection of fixed IoT devices is served by MEC services through the deployment and operation of UAVs. IoT devices have to process generated data quickly since they perform certain sensing activities. Because there aren't enough on-board computing, communication, and storage (CCS) assets, we presume that the sensor information analysis isn't done locally. Instead, we try to use resources from the UAV and already existing ground-edge communities. [3] By simultaneously optimizing task dividing decisions, UAV assignment, bandwidth for communication allocation, and calculation resource allocation at the unmanned aerial vehicle and ECs, they develop the IoT job offloading process as an irregular optimization problem to reduce the weighted average of the service postponement of all connected devices, which includes task offloading postpone and calculation delay, and UAV consumption of energy, comprised of both transmission energy and computation energy use. It is difficult to solve the optimization issue mentioned above because of the limitations and irregular functions of the objective. We use sequential convex estimates to create a successful strategy to address that difficulty. The fundamental concept of the suggested approach is to solve a series of convex smaller issues where the irregular function of objectives and restrictions is replaced with appropriate convex approximants to compute an inadequate approach to solving the original non-convex issue. By introducing the first feasible solutions, we first translate the non-convex function of objectives and constraints into appropriate convex approximants, maintaining the local initial-order behaviour of the initial non-convex issue. Next, by updating the first workable solutions to the new convex issue, they continuously compute its local optimum until the original non-convex issue finds a fixed solution. If the size of the step rule and termination condition are selected appropriately, the suggested algorithm will converge without a doubt.

2 RELATED WORKS

Large technological firms such as Siemens, Vodafone, Deutsche Telekom, and other businesses have already joined the computing at the edge industry's industrial association. Standardized MEC is being developed by the European Telecommunications Standards Institute (ETSI). To quickly and dynamically install

cutting-edge services and applications for mobile users, businesses, and vertical network segments, operators might open their mobile network edges to authorized third parties. The evolution of mobile base station iterations and the merging of IT and communications networks led to the emergence of mobile edge computing. [4] In addition to video evaluation, tracking services, the IoT, virtual reality, efficiency of regional distribution, data cached data, and other features, it will offer new vertical services to enterprise and consumer clients. To maintain mobile computing at the edge in line with 5G, ETSI published two white papers in February 2018: "Cloud the Royal Australia Navy and Mobile Edge Computation: Perfect Coupling" and "The Installation of Mobile Edge Processing in 4G Technology and Evolution to 5G." However, 5G is also being applied to the Internet of Things, including the IoT network administration system, the automotive network's intelligent allocation of resources management, the machine learning-based code distribution network, and other applications.

The offloading to other final devices has not been taken into consideration. MEC surveys have addressed methods of offloading computing and heavy data service provisioning from the end devices to the established MEC server architecture. [5] Specifically, a range of approaches have been taken by the surveys that have already been conducted to address the MEC subject area: applications and employ cases; possibilities and difficulties; security hazards and processes; calculation offloading; storing; interaction perspective; service emigration; architecture and coordination; technological advancements; and computing at the edge for the Internet of Things. MEC study is closely associated with study on computing in fog. To provide storage and computing resources, fog computing often takes into account a slightly larger set of devices than MEC that is, routers, switches, wireless access points (APs), BSs, and also devoted storage and computation nodes.

This proposes a UAV anti-jamming relaying method based on reinforcement learning to prevent significant jamming or interference with vehicle ad hoc networks. To create the conceptual and actual layout of the massive IoT system, the UAV locates the nodes in the IoT and transmits the data it gathers to the cloud. To measure how fresh the information is at the destination node, the study uses the age-of-information (AoI) metric. [6] The researchers optimize the flight path of the UAV and the energy and time of service allotments for packet transmissions to minimize the average peak AoI. A proposal is made here for an effective time-slot allocation that maximizes the utilization of frequency resources by dividing the target field into simulated hexagon cells. The researchers also address the distribution of resources and access selection in UAV-assisted IoT networks of communication, where they provide a hierarchy game system to address the shared optimization issue.

The optimization of UAV-assisted wireless IoT networks is greatly hampered by the mobility of UAVs as well as the unpredictability and dynamism of IoT systems. [7] The IoT network settings are complex and constantly changing; thus, UAVs need to be equipped with the capacity to perceive the environment and make decisions at the moment. These complex network enhancements quickly outgrow the capabilities of conventional optimization techniques. Artificial intelligence has emerged as the main cutting-edge method for IoT systems supported by drones. In particular, a lot of attention has been paid to deep learning with reinforcement (DRL), which is the combination of deep learning and reinforcement learning. DRL is becoming a very promising new technology.

This work proposes a data offloading method for secured MEC (AIMDO-SMEC) systems that combines AI and computational heuristics. To assess the traffic situation

in the MEC system, the suggested AIMDO-SMEC technique includes an efficient traffic forecasting module that utilizes Siam neural networks (SNN). [8] Additionally, MEC systems use the adaptable sample cross-entropy (ASCE) approach for data offloading. Additionally, the extreme gradient boost technique, in conjunction with the altered salp swarm method (MSSA) is used to identify and categorize cyber-attacks that occur in MEC systems. A thorough experimental analysis was conducted to examine the enhancements to the AIMDOSMEC technique. The results showed the improved results of the AIMDO-SMEC approach in terms of several variables.

3 METHODS AND MATERIALS

3.1 Scheme for proposed PSO-based deployment

The following instinct serves as our source of inspiration: A joint optimization of the UAV placement and power transmission is carried out based on (10); beginning with $K_{initial}$ UAVs; The overall count of UAVs is then increased by one if any of the generated SINRs fall short of the necessary σ_0; if the lowest SINR of every user surpasses or reaches σ_0, this process is repeated. The minimal quantity of launched UAVs, K*, is also discovered when the operation ends. In actuality, the suggested PSO-based technique will be executed offline to determine the number of necessary UAVs, their 3D hovering locations, and the transmit power allotted to each user after a computing facility in the emergency region has gathered all user positions. The suggested technique started using $K_{initial} = 1$, which requires K* iterations if K* is the limited number of UAVs. To lower the number of PSO iterations, it is therefore highly advantageous to be more precise $K_{initial}$. This rough estimate of the multi-UAV network's sum-rate downstream capacity yields a more accurate $K_{initial}$.

Our proposal is a PSO-based strategy to address the multidimensional search space for the best UAV sites. The PSO algorithm, in particular, makes use of a group of particles, each of which stands for the position vector of a UAV. [9] After that, the best-known positions of each particulate, both locally and globally, are updated by atoms that perform better than others during all of the current iterations, and this influences the particle's motion.

3.2 First initialization of K-means clustering

It is commonly known that the PSO method depends on the first-generation particles being initialized correctly. To create the horizontal locations of the first-generation particles, a starting method based on the K-means clustering algorithm is given as an alternative to random initialization. More precisely, by minimizing the subsequent error, the suggested initial strategy divides M users into K clusters.

$$\epsilon = \sum_{k=1}^{K}\sum_{m=1}^{M}\Delta_{k,m}\left\|r_m - c_k\right\|^2 \quad (1)$$

Here S_k is the user index assigned for all users clustered into the k-th group, for k = 1, 2, ... , K, and c_k is the center of the k-th clusters on the ground (also used to determine the horizontal position of the k-th UAV).

$$\Delta_{k,m} = \begin{cases} 1 & m \in c_k \\ 0 & Otherwise \end{cases} \quad (2)$$

Since $\{c_k\}$ only includes (x, y), we must add an elevation that is produced at random to each c_k in order to generate $(\bar{c}_k)$. Next, we assembled,

$$\delta_i^{(0)} = [\bar{c}_1^T, \bar{c}_2^T, \ldots\ldots \bar{c}_k^T] \quad (3)$$

Assuming i = 1, 2, … , I.

The J sets of $(\delta_i^{(0)})$ that result from the previously mentioned K-means clustering iteration are then used as the starting points for the immigrant's UAV locations that will be entered into the suggested PSO-based system. The PSO-based approach with the clustering using K-means initialization is called PSO K-means in the sequel, whereas PSO-Rand is the name given to the random initialization.

3.3 Location optimization of PSO-based UAVs

The immigrant's position vectors $(\delta_i^{(0)})$, for i = 1, 2, … , I, are the starting UAV locations in the suggested PSO scheme. Next, each of the J location vectors is updated iteratively in a way that improves the SINR values. The suggested approach optimizes the transmission power allotted to each user through the use of a SINR-balanced power management algorithm for every position in the vectors in its i_{th} generation. The following section has more details on the power transmission control method. They investigate improving the placement of many UAVs to increase the users' SINR values, in stark contrast to the PSO method, which is used to reduce the total path loss from a single UAV to all customers by maximizing the UAV location. The UAV position vector of K UAVs is shown by δ.

$$\delta = \left[e_1^T, e_2^T, \ldots\ldots e_k^T \right] \quad (4)$$

Afterward, the suggested plan continuously changes every J position vector in the direction that results in higher user SINR. To be more precise, let $(\delta^{(i)*})$ and (δ_g^*) represent the positions of vectors that correspond to the highest SINR in the i_{th} and overall generations of people, respectively. The suggested PSO algorithm first obtains the consequent SINR for every location vector (δ_g^i) in the i_{th} generation. It then ranks all J position vectors, and for j = 1, 2, … , J, according to their associated SINR value before revising (δ_g^*) and $(\delta^{(i)*})$.

3.4 The PSO algorithm uses fitness functions

By using the hybrid genetic algorithm approach, they choose a predetermined number of elements for the hybrid pool based on the hybrid frequency of each successive iteration of the algorithm. Random hybridization occurs between particles in the pool, replacing parental particles (parents) and creating a comparable number of offspring particles (children). The position of the progeny results from their father's cross.

$$Child(a) = p.\ parent_1(a) + (1 - p).\ parent_2(a) \quad (5)$$

$$Child(a) = (1 - p).\ parent_1(a) + p.\ parent_2(a) \quad (6)$$

The function uses *p*, an arbitrary number between 0 and 1, to calculate the daughter's velocity. The formula is as follows:

$$Child\ (u) = \frac{Parent_1(u) + Parent_2(u)}{Parent_1(u) + Parent_2(u)} \left| Parent_1(u) \right. \tag{7}$$

$$Child\ (u) = \frac{Parent_1(u) + Parent_2(u)}{Parent_1(u) + Parent_2(u)} \left| Parent_2(u) \right. \tag{8}$$

Each possible path is a "particle" when the aforementioned algorithm is applied to the UAV planning of routes problem. In this case, the particle's outer dimension is d, assuming d route points for control. That which is the particle fitness functions.

$$\min V = v_1 s_1 K + v_2 \frac{s_2}{d_{min}} \tag{9}$$

The weight factors in the function can be modified based on the specific task at hand. Their value dictates whether to pass over or within geographical obstacles and danger points. PSO creates a collection of randomly produced particles, and these particles regenerate by monitoring two "exceptional" values in each repetition: the first extreme value position is the optimal solution discovered by the particle, while the second excessive value point represents the optimal solution discovered by the group of particles as a whole. [10] Realizations of algorithms are:

i) Initialization: Each particle's p best connect is set to be its present location, and then we determine the corresponding someone's extreme number (that is, the physical wellness value of each particle's severe point), and the global extremely appreciate (that is, the fitness appreciate of the global extreme point) is the best one between the extremes. The selected PSO people (airway-regulate location) have a dimension of m; an initial path control point location is F, as is the velocity of the particle, which is v (usually randomly produced in the allowing range). Set g best to indicate the particle's present optimal position and note the particle's highest value.

ii) Determine the fit values of each particle and evaluate them; if the results are higher than the component's present single extreme, the particle's location will be updated, and p best is going to be defined. The particle's location will be adjusted to g best if the best single particle extreme exceeds the current global extreme. We then log the number of particles and update the global extreme.

iii) Update the location and speed of every single particle using the shrinking factor equation.

iv) Compare the value of adaptation of every single particle with the optimal location it passed, and if it is better, establish that as the optimal location.

v) Update g best by comparing the present values of p best and g best;

vi) The hybrid chance determines how many particles are chosen to go into the hybrid pool. The particles in the pool will randomly hybridize with one another to produce an identical number of daughter particles. The function determines the daughter's position and velocity. Maintain the same p and g.

vii) Verify that the ending requirements are met. If the present iteration has reached the predetermined optimum number, it should finish, produce the best answer, or go on. Otherwise, it should stop.

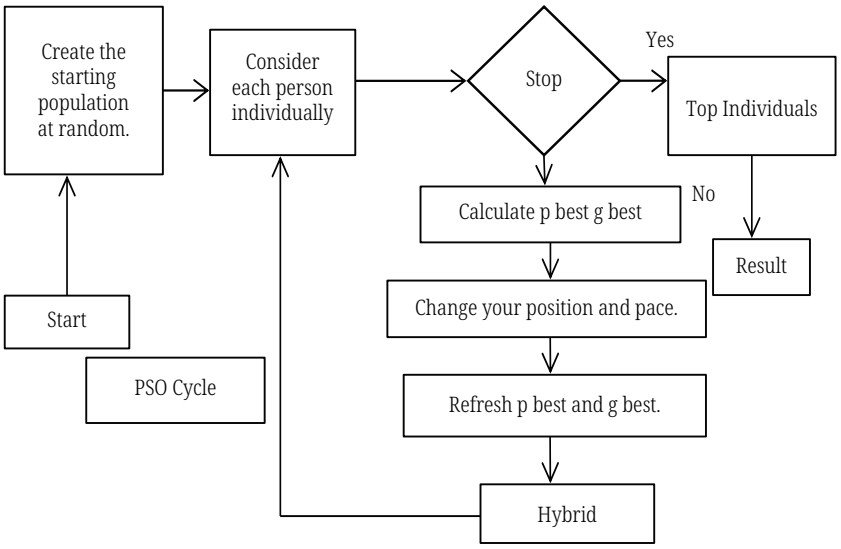

Fig. 3. An improved flow chart for the particle swarm algorithm

The optimization issue can be challenging to resolve using conventional methods due to its immense complexity and NP-hardness. Nonetheless, DRL is a useful approach to resolving this issue. In this part, we first describe the compute offloading problem. [11] By figuring out the cars' offloading decisions, we want to optimize the long-term return. Every agent is associated with a vehicle, and they all interact with the environment through a policy that outlines the choices the agent makes at each stage. To overcome the compute offloading issue and reduce the unpredictability of the multi-vehicle environment, we thus make use of the recently developed multi-agent deep learning with reinforcement.

3.5 DRL for vehicles networks

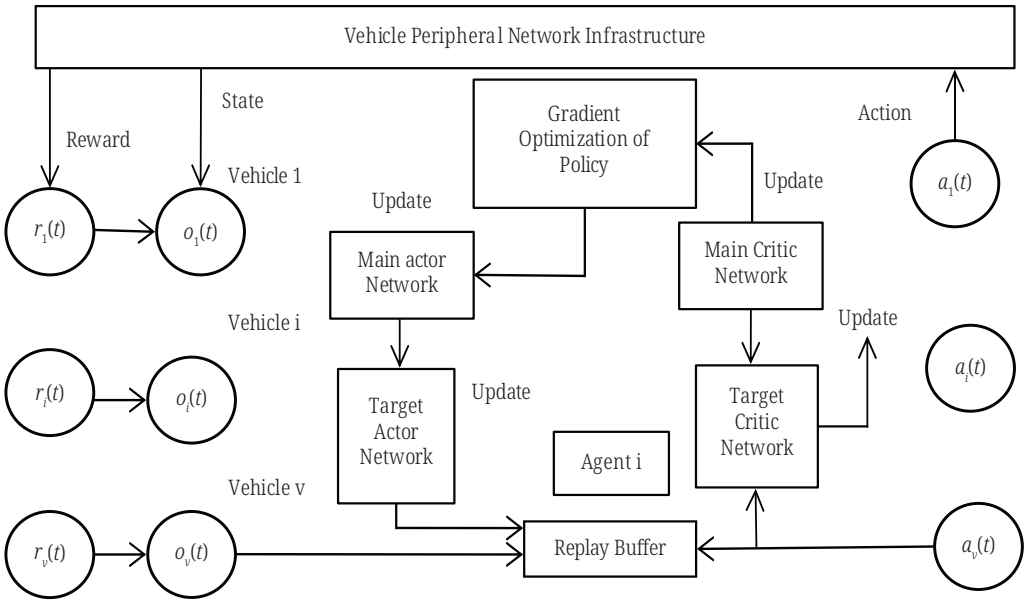

Fig. 4. Architecture for the automotive network's multi-agent DRL-based compute offloading

In the multi-agent model, we will include the system's state, behavior, and reward. Figure 4 displays the architecture for the vehicular network's multi-agent DRL-based compute offloading. The MDRCO learning procedure involves the distributed execution of the intended actor networks and centralized training of the primary actor and major critic networks. To obtain the automobile's state (o_j), every agent i follows the primary operator network (π_j). The vehicle gets the reward (r_j) after doing the matching action (a_j). After that, the car enters the following state (o_j). All of the cars' joint states (s), shared actions (a), shared rewards (r), and future states (s) are stored in the replayed buffers D by the data center. The three steps of network instruction are as follows: initially, upgrade the primary critic network. 2. Refresh the primary actor network. Then update the target actor network and target critic network last.

4 RESULT ANALYSIS AND DISCUSSION

4.1 An algorithm for DRL-based offloading

The offloading technique that we present in this part is based on DRL and allows all mobile devices to arrive at decisions about offloading in a distributed way. Deep Q-learning is the basis of this method. Since deep learning using Q-learning constitutes a model-free method, the suggested algorithm can handle the complex configuration and mobile phone interface without requiring a previous understanding of the dynamics of system components and interactions. In the meantime, the system's possibly vast state space can be managed by the proposed approach. [12] Under the suggested approach, every mobile device is supposed to learn a mapping between every state-action combination and a Q-value that represents the state-action pair's anticipated long-term cost. A network of neural networks decides the mapping. To minimize its predicted long-term expenses, every gadget can choose the action that produces the lowest Q-value in its current condition based on the map. The neural network's structure and the DRL-based method are shown below, in that order.

4.2 Performance evaluation

One thousand by one thousand square meters, or one hundred MDs, make up the simulation area. An MD's processor operates at a frequency of one gigahertz, and the input data size for processing jobs is spread out randomly within the interval [15, 25] Mbit. Here, ten unmanned aerial vehicles, or UAVs, are stationed, with the capacity for each drone to provide computational services to several MDs. Within the range of [3.5, 4.5] GHz, UAVs' CPU frequencies are dispersed arbitrarily. Table 2 displays more parameter configurations. The four types that follow are taken into consideration, with varying methods of MD transportation based on the values of the parameters.

Table 2. Parameter settings

Parameter	Value
B	100 Mhz
H	25 m
P	1 watt
D	100 cycles
Nmax	10

Type 1: In this type, an enormous number of MDs are produced. In particular, this local area is where around 92% of MDs gather, with the remaining MDs being dispersed at random throughout other places.

Type 2: In this type, MDs are similarly gathered in a small region, but only half of the MDs are gathered there; the remaining MDs are dispersed at random over various areas.

Type 3: In this type, MDs are gathered in two neighborhood districts. In other words, half of the MDs are gathered in one region, another half is formed in another region, and the other MDs are dispersed at random throughout the remaining areas.

Type 4: In this type, MDs are dispersed randomly and do not congregate in any particular local location.

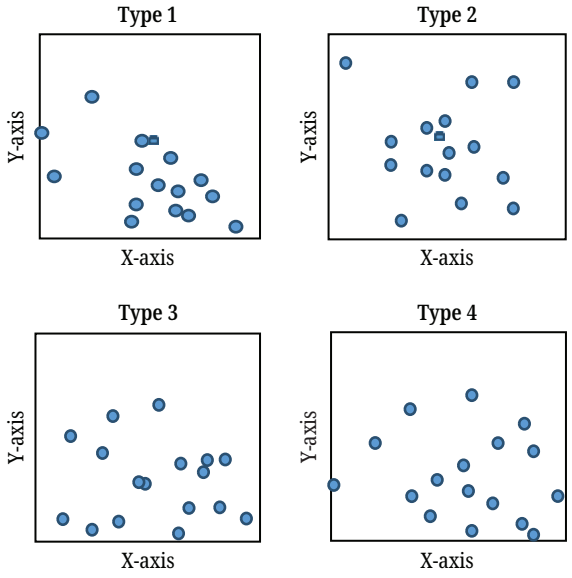

Fig. 5. Various MD distributions under different contexts

4.3 Baseline techniques

The following three base approaches are modified to match the provided problem to test the viability and efficacy of the suggested PSO-G [13].

Random greedy algorithm: To solve the optimization problems of UAV placement and computational offloading, respectively, the random approach and the greedy method are used. Random G treats the setting up of UAVs and computation offloading as two entirely separate procedures and does not take into account the effects of MD locations and computing task resource needs on optimization outcomes.

K-means greedy algorithm: The optimization problems of UAV distribution and computing offloading are tackled by the K-means and pessimistic methods, respectively. The K-means-G algorithm treats the deployment of UAVs and computation offloading as two separate procedures.

PSO–greedy algorithm: To solve the efficiency problems of UAV deployment and compute offloading, accordingly, the PSO and greedy methods are used. Computation offloading and UAV deployment are concurrently taken into consideration in the PSO-G. To assess the results, the PSO-G, K-means-G, and random-G are evaluated. Figure 6 presents the findings.

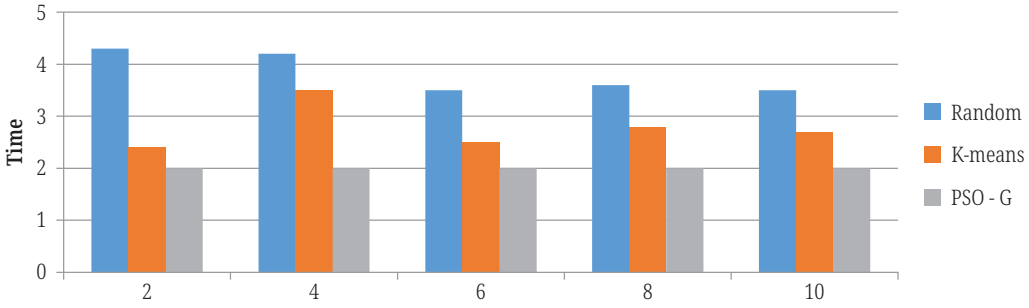

Fig. 6. Performance evaluation

When compared to Types 1 and 2, the K-means-G's average task reaction time essentially stayed the same, but the PSO-Gs significantly increased. According to Figure 7, all of these approaches' average task reaction times decrease with an increase in UAV count. This is because the likelihood of computing jobs being transferred from MDs to UAVs for completion increases with the number of UAVs deployed, as does the reduction in distance between MDs and UAVs. In all cases, the PSO-G can perform better than the other three methods.

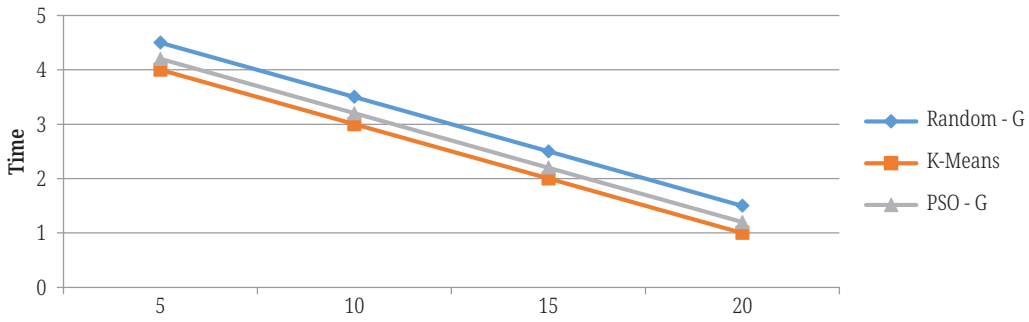

Fig. 7. Total number of UAV's

5 CONCLUSION

Mobile edge computing offers mobile end-user applications an improved quality of experience (QoE) and a short-latency alternative to traditional centralized cloud computing. Implementations of MEC infrastructure are becoming increasingly necessary as the number of applications with high processing requirements and vital missions rises. We have outlined the four stages of MEC infrastructure implementation and suggested high-level actions for deploying an integrated end-to-end MEC-UAV architecture. Our proposal involves the appropriate deployment of a UAV to enable the distribution of MEC services to a group of static IoT devices in areas where grounded signal obstruction and shadow prevent IoT devices from accessing the current ECs. Through aerial-to-ground relations, the UAV and ECs in our system work together to supply MEC functions to the IoT devices. To minimize the weighted sum of the connection delay of all IoT devices and UAV use of energy, we have designed a non-convex optimization challenge that involves optimizing UAV position, communication, computing allocations, and job splitting choices in tandem. According to the simulation results, the enhanced PSO algorithm can successfully tackle complicated route planning problems for UAVs and eliminate the basic PSO algorithm's tendency to become trapped in local minimums. The average task

reaction time can be efficiently optimized by the PSO-G; however, when combined with the greedy algorithm, it may enter a local optimum due to its low effectiveness in training.

6 REFERENCES

[1] L. A. Haibeh, M. C. Yagoub, and A. Jarray, "A survey on mobile edge computing infrastructure: Design, resource management, and optimization approaches," *IEEE Access*, vol. 10, pp. 27591–27610, 2022. https://doi.org/10.1109/ACCESS.2022.3152787

[2] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2018. https://doi.org/10.1109/JIOT.2018.2887086

[3] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3147–3159, 2020. https://doi.org/10.1109/JIOT.2020.2965898

[4] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020. https://doi.org/10.1109/ACCESS.2020.2991734

[5] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. Fitzek, "Device-enhanced MEC: Multi-Access Edge Computing (MEC) aided by end device computation and caching: A survey," *IEEE Access*, vol. 7, pp. 166079–166108, 2019. https://doi.org/10.1109/ACCESS.2019.2953172

[6] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10187–10200, 2019. https://doi.org/10.1109/TVT.2019.2935877

[7] Y. Yu, J. Tang, J. Huang, X. Zhang, D. K. C. So, and K. K. Wong, "Multi-objective optimization for UAV-assisted wireless powered IoT networks based on extended DDPG algorithm," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 6361–6374, 2021. https://doi.org/10.1109/TCOMM.2021.3089476

[8] F. Alrowais *et al.*, "Artificial intelligence based data offloading technique for secure MEC systems," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 2783–2795, 2022. https://doi.org/10.32604/cmc.2022.025204

[9] W. Liu, G. Niu, Q. Cao, M. O. Pun, and J. Chen, "Particle swarm optimization for interference-limited unmanned aerial vehicle-assisted networks," *IEEE Access*, vol. 8, pp. 174342–174352, 2020. https://doi.org/10.1109/ACCESS.2020.3025897

[10] Q. Geng and Z. Zhao, "A kind of route planning method for UAV based on improved PSO algorithm," in *25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 2328–2331. https://doi.org/10.1109/CCDC.2013.6561326

[11] X. Zhu, Y. Luo, A. Liu, M. Z. A. Bhuiyan, and S. Zhang, "Multiagent deep reinforcement learning for vehicular computation offloading in IoT," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9763–9773, 2020. https://doi.org/10.1109/JIOT.2020.3040768

[12] M. Tang and V. W. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 1985–1997, 2020. https://doi.org/10.1109/TMC.2020.3036871

[13] Z. Chen, H. Zheng, J. Zhang, X. Zheng, and C. Rong, "Joint computation offloading and deployment optimization in multi-UAV-enabled MEC systems," *Peer-to-Peer Networking and Applications*, vol. 15, pp. 194–205, 2022. https://doi.org/10.1007/s12083-021-01245-9

7 AUTHORS

Dr. N. Vijayalakshmi is an Assistant Professor (Sr.G), Programme Coordinator Data Science, at the Department of Computer Science and Applications, SRMIST, Ramapuram. She completed her MCA in 2007 and her doctorate at Anna University in 2018. Her area of interest is image processing and machine learning. She published 25 research articles in reputed journals and conference proceedings. She has international and national patents (E-mail: vijayaln@srmist.edu.in).

Sagar Gulati is a resilient academic leader with over a decade of experience in academics and research, policy formulation and execution, academic administration and control, projects management, resources optimization, outcome based education and business development. Presently, he is Director, School of CS and IT Jain (deemed-to-be university). He has been a strategic adviser to the World Bank sponsored STRIVE project for capacity development of apprentices in Haryana; dean of computing at the college of Engineering Roorkee; director of technology education research integrated institutions Kurukshetra and head TERii technology business incubator, recognized by the ministry of MSME, Govt. of India. He is also Country Representative–India, for Lincoln University College, Malaysia. He is also in the think tank and on the Board of Governors of the Inventivepreneur Chamber of Commerce & Industry, New Delhi, and on the Member, Board of Advisors of the IREU Universe, Bangalore. He is the recipient of the Eminent Engineering Personality Award by the Institution of Engineers (India) and appreciation awards from His Excellency the Governor of Odisha, Prof. Ganeshi Lal, and His Excellency the Governor of Himachal Pradesh, Acharya Devvrat, for his exemplary contributions in the higher education domain. His area of research is distributed systems and artificial intelligence; he has a number of research publications, Industry-sponsored consultancy projects, and government-sponsored R&D projects under his credit (E-mail: sagar.gulati@jainuniversity.ac.in).

Dr. B. Ben Sujin is a distinguished faculty member and researcher in computer engineering at the University of Technology and Applied Sciences (UTAS) in Nizwa, Sultanate of Oman. As an active member of IEEE and the counselor for the university's IEEE Student Branch, Dr. Sujin has made significant contributions to his field. In recognition of his outstanding research efforts, Dr. Sujin received the Best Active Researcher Award from UTAS in 2018 and has been honored multiple times with the Best Faculty Award. He is a prolific author, having written numerous research papers and books, and presented his work at various international conferences. Dr. Sujin has successfully secured numerous funded projects from the Research Council of Oman and the Institution of Engineers, highlighting his expertise and leadership in research. His primary areas of expertise include robotics, artificial intelligence (AI), and the Internet of Things (IoT). A veteran in his field, Dr. Sujin is passionate about educating the next generation of engineers. He has conducted numerous seminars and webinars, inspiring and enlightening students in the fields of robotics, IoT, AI, and other emerging topics in computer engineering (E-mail: bennet@utas.edu.om).

Dr. B. Madhav Rao is working as an Associate Professor in the department of CSE, Sir C R Reddy College of Engineering, Eluru, A.P. He has received his Ph.D. in Computer Science from Rayalamaseema University, Kurnool, and M. Tech (CST) from Andhra University, Visakhapatnam, A.P., India. His research interest includes, deep learning, network security and bio-Informatics (E-mail: madhavaraob@sircrrengg.ac.in).

Dr. K. Kiran Kumar is working as a professor at K. L. Deemed University. He had 25+ years of teaching experience. He had good knowledge of operating system, artificial intelligence, software engineering, web application development, python, java, IoT, data mining, machine learning and deep learning. He had published 43 papers in various international journals and conferences. Six scholars are working under his guidance. He guided various student projects at the academic level. He had given guest lectures in various institutions and attended various national and international seminars. He is working as the professor in charge of academics. He designed the syllabus for various courses. His research area includes artificial intelligence, machine learning, deep learning, block chain, IoT, and data mining (E-mail: kiran5434@kluniversity.in).