International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 4 | Pages 909-919|e-ISSN: 2347-470X

A Multi-Agent Deep Reinforcement Learning Framework for MEC Resource Allocation in 5G Networks

Meghamala Y[1*], Pulipati John Paul[2], and M Aravind Kumar[3]
[1]Department of Electronics and Communication Engineering, Bharatiya Engineering Science & Technology Innovation University (BESTIU), Andhra Pradesh, India; Email: meghamala7@gmail.com
[2]Department of Electronics and Communication Engineering, Ellenki College of Engineering and Technology, Patancheru, Telangana, India; Email: jppulipati@yahoo.com
[3]Department of Electronics and Communication Engineering, West Godavari Institute of Science and Engineering, Prakashraopalem, East Godavari, Andhra Pradesh, India; Email: drmaravindkumar@gmail.com

*Correspondence:** *Meghamala Y; Email: meghamala7@gmail.com*

ABSTRACT- This paper introduces a multi-agent Deep Reinforcement Learning (DRL)-based model of allocating resources in 5G MEC networks based on the Soft Actor-Critic (SAC) algorithm and the hierarchical MATD3/TD2PG-based actor-critic network. The model distributes sub-channels, power of transmission and MEC computing resources with taking into account user mobility and isolation of the slices. The Python simulation is provided with a Manhattan 5G environment comprising of four interconnected gNodeBs, 5 densities of users (327, 499, 596, 930 and 1088 users), and two MEC classes of service (security and entertainment) with predefined bandwidth, memory, and processing requirements. It is assessed against three baselines: Greedy, Best-fit and Worst-fit allocation strategies in three measures; number of services served, services blocked and services denied. Findings indicate that SAC-based allocation improves the number of services served by 8-14, blocked by 15-22 and denied services by 18-20, respectively, with respect to user density. The advantages of these results support that the suggested multi-agent model, which is SAC-based, offers a measurable performance increase in the given dynamic traffic and heterogeneous service conditions.

Keywords: 5G, MEC, Resource Allocation, DRL, SAC Model.

ARTICLE INFORMATION
Author(s): Meghamala Y, Pulipati John Paul, and M Aravind Kumar;
Received: 31/08/2025; **Accepted:** 26/11/2025; **Published:** 30/12/2025;
E- ISSN: 2347-470X;
Paper Id: IJEER 3108A24;
Citation: 10.37391/ijeer.130431
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-13/ijeer-130431.html

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

The creative ways to allocate resources efficiently and dynamically for 5G networks can be deployed quickly. While existing approaches do work to a certain degree, they are not scalable or adaptable enough to deal with complicated, ever-changing settings in real time [1][2]. Wireless network resource allocation is currently a difficult topic, but it is about to become much more so with the advent of more advanced mobile networks like 5G and beyond, as well as the multitude of devices and new use cases that will need their support [3] [4]. Mobile Edge Computing (MEC) is a rapidly evolving technology that allows end devices to directly offload computationally heavy queries to servers located at the edge of wireless networks [5]. The quality of the user experience may be greatly enhanced by offloading, which substantially improves performance metrics such as execution latency and energy usage. However, effective offloading remains challenging due to the dynamism and unpredictability of computation request arrivals, device energy constraints, varying radio environments, and MEC server resource limitations [6].

In many scenarios, a large number of devices can lead to network overload, which in turn reduces performance. Furthermore, devices must carefully regulate both energy and bandwidth usage [7][8]. Wireless personal area networks, in particular, have limited bandwidth; therefore, devices must be selective about what data they transmit and receive [9][10]. Link quality measured in terms of latency and reliability is directly impacted by how well resource allocation and planning techniques function in Cellular Vehicle-to-Everything (C-V2X) communication. However, the continuous mobility of vehicles makes it impractical to maintain a single centralized coordinator to manage real-time resource allocation [11][12].

The 5G cellular networks are also designed to support a variety of smart applications requiring substantial bandwidth. Reusing frequencies through Device-to-Device (D2D) communication is one possible method to enhance 5G throughput [13]. Similarly, adaptive buffering techniques based on HTTP streaming enable media players to dynamically adjust bitrates depending on network performance [14]. In addition, the deployment of small cells in modern Heterogeneous Networks (HetNets) improves network capacity, although it introduces complex interference challenges between macro and small cells [15][16].

Since 5G was introduced, network traffic patterns, QoS requirements, and scalability challenges have shifted due to the massive growth in connected devices and applications [17][18]. Network slicing has emerged as a promising solution, enabling operators to create multiple virtual slices on a shared physical infrastructure and allocate resources according to slice-specific requirements [19][20]. Heterogeneous networks (HetNets), combined with techniques such as Downlink/Uplink Decoupling (DUDe), further improve resource efficiency in 5G MIMO networks [21]. However, ultra-dense 5G environments, storage/processing constraints, and resource heterogeneity make real-time service provisioning particularly difficult [22].

Finally, the rise of IoT has increased the importance of intelligent resource management strategies in 5G. While IoT enhances connectivity, it also introduces vulnerabilities in crisis scenarios where devices may face energy shortages or cyberattacks [23]. Therefore, advanced approaches leveraging Artificial Intelligence (AI), Deep Reinforcement Learning (DRL), and game theory are increasingly being explored for dynamic and energy-efficient resource allocation in 5G networks [24].

This paper makes the following focused contributions:

- *A Multi-Agent DRL Architecture:* We design a hierarchical system combining Soft Actor–Critic (SAC), MATD3, and TD2PG to jointly optimize sub-channel allocation, power distribution, and MEC resource assignment under slice isolation constraints.
- *Integration of Mobility and Service Duration:* The proposed policy explicitly incorporates user mobility (Random Waypoint model) and differentiated service durations for security and entertainment tasks, improving decision accuracy for long-running MEC workloads.
- *Realistic 5G MEC Simulation Framework:* Experiments are conducted in a Manhattan-grid 5G scenario with four gNodeBs, five user densities (327–1088 users), and MEC resource configurations taken directly from real service profiles.
- *Quantitative Performance Analysis Using Three Key Metrics:* Using services served, services blocked, and services denied as evaluation indicators, the method is compared against Greedy, Best-fit, and Worst-fit baselines.
- *Clear Numerical Improvements Demonstrated:* The proposed method yields 8–14% higher service completion, 15–22% fewer blocked services, and 18–20% fewer denied services across all user densities.

What follows is an outline of the rest of the paper: *Section 2* details work that are relevant to the topic. *Section 3* explains the assumptions and model of the system. *Section 4* details the planned procedure. In addition, the suggested scheme's performance analysis is presented in *section 5*. The numerical outcomes of the suggested strategy are shown in *section 6*. *Section 7* concludes with some last thoughts.

2. RELATED WORK

The authors in [25] propose an energy-aware mode-selection system for Device-to-Device (D2D) resource allocation in 5G networks, where D2D and traditional cellular users coexist. To mitigate interference caused by uplink transmit power, users are categorized into three groups based on distance from the base station, and spectrum allocation is performed using the Hungarian algorithm.

A hybrid machine learning framework, termed Dynamic Resource Allocator using RL-CNN (DRARLCNN), is proposed in [26]. It integrates CNN-based feature extraction with reinforcement learning for decision-making, trained using the "5G Resource Allocation Dataset" and tested in a simulated environment built on Python, TensorFlow, and OpenAI Gym. Results show superior performance compared to existing methods, with reduced latency and improved allocation efficiency.

In [27], a Multi-Agent Reinforcement Learning (MARL) based decentralized allocation approach is presented, where each User Equipment (UE) independently allocates resources while jointly learning a shared policy. The study evaluates Independent Learners (ILs) and Value Function Factorization (VFF) using QTRAN-based centralized training with decentralized execution. Results demonstrate that MARL enables effective distributed resource allocation, improving throughput while satisfying per-user QoS.

The work in [28] addresses dynamic optimization in 5G MEC heterogeneous networks with energy-harvesting mobile devices. Using queuing theory, the authors analyze static and dynamic subchannels separately, applying a Simulated Annealing Genetic Algorithm (SAGA) with Lyapunov optimization to balance computation offloading and resource sharing. An energy-efficient allocation strategy for D2D communication in wide-area 5G networks is developed in [29]. The scheme reduces energy consumption, extends device lifetime, and improves communication efficiency, thereby lowering the overall environmental footprint.

A comprehensive review in [30] explores deep reinforcement learning (DRL) for resource allocation in 5G Cloud-RAN (C-RAN). The paper highlights the potential of DRL to autonomously learn complex policies while discussing issues such as scalability, fairness, and convergence. Finally, [31] examines radio resource allocation in C-V2X networks using a decentralized multi-agent actor-critic framework. Two variants Independent Actor-Critic (IAC) and Shared Experience Actor-Critic (SEAC) are compared. Simulation results in high-density vehicular networks indicate 15-20% improvements in reliability over baseline methods.

3. PROPOSED WORK

In this part, we provide a design scheme for a dynamic optimal resource allocation method for the eMBB scenario and an URLLC scenario, respectively, based on the various network business needs indicated earlier.

3.1. eMBB

Prior to explaining the model for allocating resources, few variables need to be specified: The flow of electricity traffic rates of the tenant networking slice is represented by B^{cu}, B^{pr} is the bandwidth that has been pre allocated to it based on the statistical findings of user demand. Due to this, B^{cu}/B^{pr} is represented by θ and signifies the resource utilization ratio of the present network slice. We also use $\hat{y}^H$ and $\hat{y}^L$ to denote the maximum and minimum values for the resource usage ratio, correspondingly. All things considered; the following usage states of network slices are possible:

Wasted: θ≤η_L;
Feasible: η_L≤θ≤η_H;
Congested: θ≥η_H;

For optimal user experience, bandwidth allocation to a tenant network slice should be increased when the network enters a congested state and reduced when it is in a wasted state. In the feasible state, where utilization is within acceptable limits, adjustments are unnecessary. In eMBB scenarios, the high throughput demands require substantial resource allocation and the maintenance of a stable utilization ratio. Deviations from this ratio can lead to network congestion and degraded QoE if too high, or resource wastage and increased costs if too low.

To address this, the Admission Control mechanism is applied. Specifically, this work employs the Worst-Case Admission Control (WAC) method, which ensures that while the resource utilization ratio remains within the defined range, resource allocations are not dynamically altered. As shown in *figure 2*, the parameter r_u/c_u s_o represents the predetermined proportional threshold of a slice and denotes the percentage of resources requested by a new user.

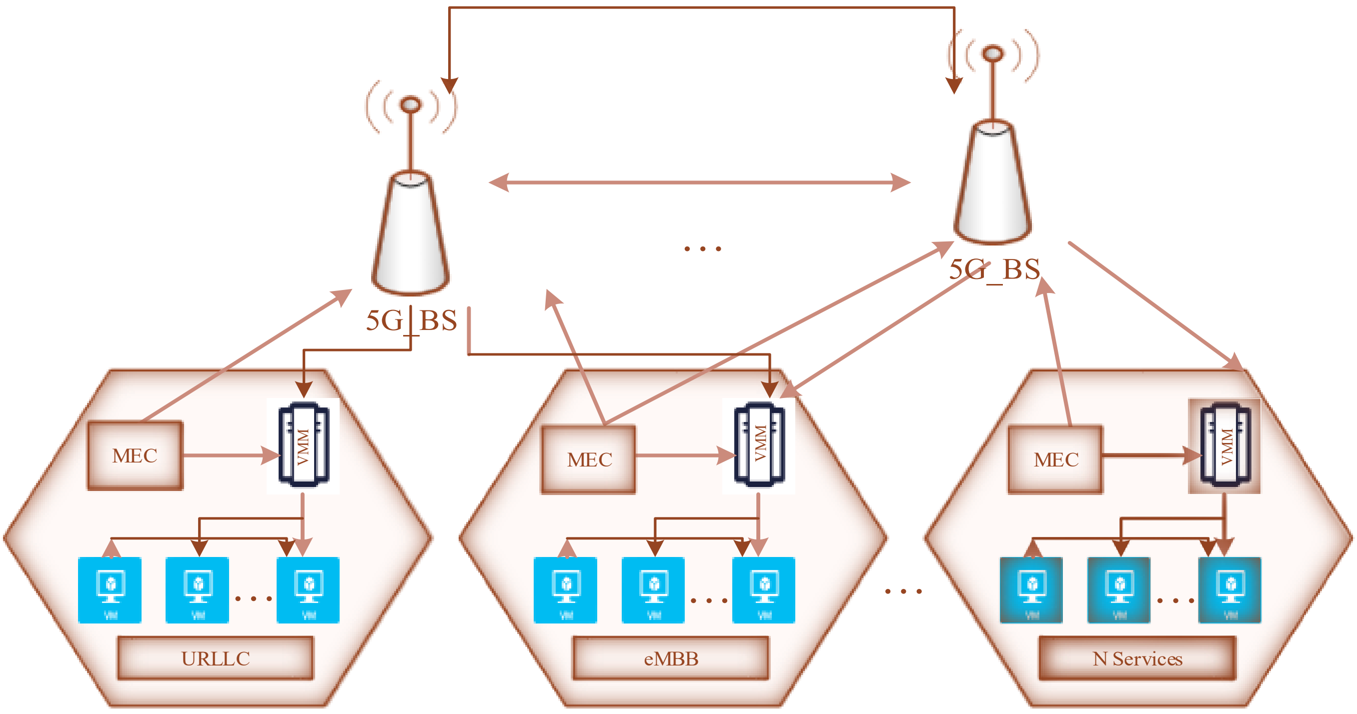

Figure 1. 5G Networks Model

As a result, the resource consumption proportion within network slice *i* is given as:

$$\frac{\sum_{j=1}^{m} b_j}{B_i^{ad}} \tag{1}$$

where b_j represents the current traffic rate for j^{th} slice. Once the resource allocation plan for the tenant network segments is complete, optimization of the resource usage ratio results in:

$$\max\bar{\theta} = \frac{1}{n}\sum_{i=1}^{n} \theta_i \tag{2}$$

Although monetary benefits to adjusting network resources exist, costs incurred (such as transmission losses or update times for tenant network slices) need to be taken into consideration. Upon changing the unit bandwidth for network i^{th} slice, q_i is used to express the cost sustained. It has to be made sure that the adjusted bandwidth does not incur exorbitant costs, thus

$$\min Q = \sum_{i=1}^{n} q_i \left|B_i^{ad} - B_i^{pr}\right| \tag{3}$$

Keeping in mind that maximization and cost reduction as prerequisites for the a fore mentioned objective functions, maximizing the normalized parameter γ, results in the following goal function:

$$\max\gamma = \frac{\frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{m} b_j}{B_i^{fu}}}{\sum_{i=1}^{n} q_i|B_i^{ad}-B_i^{pr}} \qquad (4)$$

considering the constraints:

$$\begin{cases} 0 \le B_i^{ad} \le B^{\text{total}} & \forall i \in [1,n] \\ \eta_L \le \theta_i \le \eta_H & \forall i \in [1,n] \\ \sum_{i=1}^{n} B_i^{\text{ad}} \le B^{\text{total}} \end{cases} \qquad (5)$$

3.2.URLLC

Future applications of machine-to-machine based services can be utilized in areas such as industrial control, smart transportation, environmental monitoring, *etc.*; the Ultra DRL along with Low Latency Communication (URLLC) situation aims to address the stringent latency and reliability requirements of these applications.

In addition to meeting user requests, lower latency accuracy may deliver larger rewards. In the tenant's network slice, p denotes the profits generated per unit bandwidth, it can be stated by the following formula:

$$p = p_0 - \xi_{td} \cdot t_d \qquad (6)$$

Here, p_0 is the base value of profits, ξ_{td} is the delay time and t_d is the penalty factor. In addition, the total profit for any slice P may be stated as:

$$P = p \cdot B^{cu} \qquad (7)$$

since the network load and by extension, the delay time t_d are both affected by changes in the tenant network slice's current traffic rate:

$$t_d = \tau_{bd} \cdot B^{cu} \qquad (8)$$

where τ_{bd} factors into the load penalty. For the current delay time, t_{cu}, and for the delay time prior to load increase, we use t_{pr}. While $t_{cu} > t_{pr}$, The profit of the tenant network's slice will be decreased, as shown in *equation (5)*. The objective is to aim at optimization of the profit obtained by the adjustable tenant networks slice, which is the adjusted delay time, t_{ad}.

$$\max P_{\text{total}} = (p_0 - \xi_{td} \cdot t_{ad}) \cdot B^{cu} \qquad (9)$$

But the expense of resource scheduling cannot be overlooked. Assuming that d_i is required to decrease the unit delay time on slice i, the secondary objective remains to minimize the cost of delay reduction, which is:

$$\min D = \sum_{i=1}^{n} d_i|t_{ad} - t_{cu}| \qquad (10)$$

It is necessary to maximize the normalized variable ρ, which allows for the characterization of the objective function in a similar way.

$$\max\rho = \frac{\sum_{i=1}^{n} B_i^{cu} \cdot \left(p_0 - \xi_{td} \cdot \sum_{j=1}^{m} t_j\right)}{\sum_{i=1}^{n} d_i|t_{ad} - t_{cu}|} \qquad (11)$$

where t_j is the tenant network's *n*th slice adjusted delay time for *j*th user on *i*th slice. considering the constraints:

$$\begin{cases} 0 < t_{ad} < t_{cu} \\ d < \xi_{td} \cdot B^{cu} \\ \sum_{j=1}^{m} t_j < \frac{p_0}{\xi_{td}} \ \forall \text{ slice } i \in [1,n] \end{cases} \qquad (12)$$

3.3. Network Scenario

We imagine a 5G world where every mobile user (including cars, other mobile users, and other gadgets) u_e has a unique identity $e \in [1,w]$, where the upper limit on users is represented by w. The computing capabilities of a particular mobile user u_e would not be able to handle the processing of a service that they could want at any given time. So, for MEC service processing, the device communicates with a 5G network infrastructure controller node CN *via* a request message. We take MEC $m_k(k \in [1,o]$, A mobile edge computing (MEC) cluster, where o is the maximum number of nodes, is a collection of devices that have similar preferences and might pool their resources to provide 5G network nodes with access to more resources. More specifically, a controller node CN may gather and manage idle computational resources, including processing or storage, from a specific mobile node u_e.

In this context, a given mobile node u_e could increase its capabilities by using the available resources of MEC m_k, while other entities lend their resources to MEC m_k.Therefore, MEC m_k could provide services s_a ($\in [1,q]$ where m is the maximum of the number of services) up to w_{lim} mobile users at the network edge. *Figure 1* shows the scenario in which the DRL can be deployed in the controller node to manage the resources coming from the urban environment composed of mobile nodes connected through the 5G network infrastructure. The controller node is a centralized entity that has a global view of each MEC iteration and all users to allow better allocation.

Here, a mobile node u_e, which may be circulating throughout an urban area, could communicate with the controller node CN to request a service s_a, which could include things like tracking traffic or entertainment options. Processing, storage, and runtime are computing resources that this service s_a, needs in order to fulfill user requests effectively. Here, the DRL resource allocation method shown in *Figure 2* is used by the controller node CN to decide when and where to assign the service s_a, on a specific MEC m_k after receiving the request. To emphasize, in order to determine whether MEC m_k has the necessary resources, the controller node CN has a bird's-eye view of all services and node statuses r_{serv} to allocate an assumed service s_a.

A service s_a may be allocated using resources made accessible by MEC m_k, as requested through users. We assume that each MEC can handle a maximum of $w_{\text{"}lim\text{"}}$ users and $q_{\text{"}serv\text{"}}$ services in order to address the issue of resource allocation for a certain service s_a. Lastly, in order to continue efficiently fulfilling the user's request, the controller node CN may have to move a specific service s_a onto a different MEC device. This might be attributed to factors such as user mobility, resource availability, or having to provide load balancing.

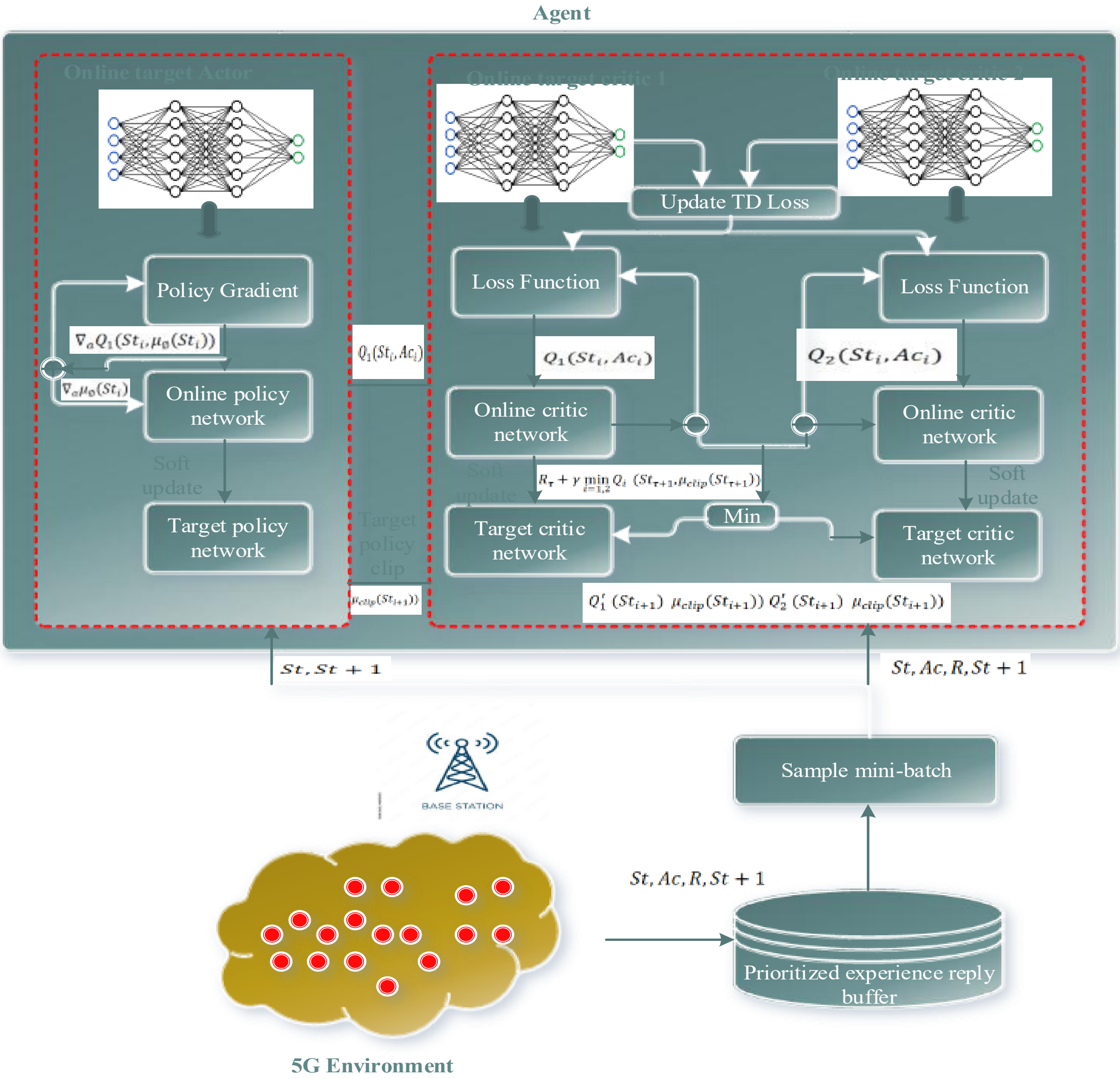

Figure 2. DRL based Resource Allocation in 5G Networks

There has been extensive usage of reinforcement learning because it provides a better system for allocating a specific real-time computer resource. To be sure, the algorithm's performance is heavily dependent on the architecture and reinforcement learning components used. Various issues call for various reinforcement learning architectures. Given the difficulties of the suggested model, we use MATD3 to distribute resources at the highest level and DCTD3 to distribute resources at the lowest level. In the real world, we train a single agent for every slice to distribute resources to users, rather than using many agents since we need to guarantee mutual isolation along with security across slices.

3.4. Multi-Agent Actor and Critic Networks

The RCs in the proposed system constantly update its routing along with accident detection algorithms based on what they learn about the network's present condition. The three separate components that make up RC in our work are the Data Collection Mechanism (D_m), the Trajectory Component (T_m), and the Routing Module (R_m). Our proposed method, TD2PG, is a twin-delayed deterministic policy gradient for the purpose of intellectual learning. For better routing and accident dispersion, the suggested TD2PG algorithm learns each vehicle's route. One of TD2PG's characteristics is its actor critic architecture, which allows the critic network to take a state St and an action Ac as inputs while offering the value of Q as an output Q (St, Ac). The current state of the network is denoted by St, and the vehicle's trajectory forecast is AC. Data about the present condition is gathered by the RCs.

The action $a_{t,1} = \left(a_{n,k}(t)\right)_{1\times(3-K)}$ at time t, the first agent takes action concerning the allocation of sub-channels for three slices. We allocate it to the n^* slice that has the highest value for every k sub-channels., $n^* = \arg\max_n \left(a_{n,k}(t)\right)$-the value of $v_{n^*,k}$ is set to 1, the other slice's $v_{(n,k)}$ is initialized to zero at time *t+1*. Hence, the sub-channel allocation V(t+1) of the two agents' states is affected by the action $a_{(t,1)}$.

The action $\mathrm{a}_{t,2} = (a_n(t))_{1\times 3}$ to the second agent with respect to the distribution of power among the three slices at time t). The power action bound at time *t* in the upper-level optimizer is denoted by ∇p_1, which may be increased or decreased at each time-step. Using the second agent's actions and *equation (6)*, we can calculate the power allocation *P(t+1)*:

$$\mathrm{P}(t+1) = \mathrm{P}(t) + \mathrm{a}_{t,2} * \nabla p_1 \tag{13}$$

That is why the agents' and the environment's state *P(t+1)* will be affected by the actions $\mathrm{a}_{(t,2)}$. Two existing networks of critics $Q_1(\mathrm{S}_{t,m}, \mathrm{A}_t \mid \theta^{Q}_{m,1})$ and $Q_2(\mathrm{S}_{t,m}, \mathrm{A}_t \mid \theta^{Q}_{m,2})$ with weights $\theta^{Q}_{m,1}, \theta^{Q}_{m,2}$ are initialized at random and used to predict the *m^th^* agent's Q-function in MATD3. Furthermore, we set up a single current actor network from the outset $\mu(\mathrm{S}_{t,m} \mid \theta^{\mu}_{m})$ with weights $\theta^{Q}_{m,1}$.

On line 2 of the initial algorithm for every *m*th agent, where $A_t = (\mathrm{a}_{t,1}, \mathrm{a}_{t,2})$. Applying the deterministic policy gradient, the current actor network selects a deterministic action according to the state $\mathrm{S}_{t,m}$ at time *t*. At time *t*, the *m*th agent's action $\mathrm{a}_{t,m}$ may be expressed as:

$$\mathrm{a}_{t,m} = \pi(\mathrm{S}_{t,m}) = \mu(\mathrm{S}_{t,m} \mid \theta^{\mu}_{m}) + \epsilon_1. \tag{14}$$

where $\epsilon_1 \sim \mathcal{N}(0, \sigma_1^2(t))$ constitutes typical random noise. More motions may be explored with it. With more training epochs, the noise's variance drops, which means, $\sigma_1^2(t+1) = \eta\sigma_1^2(t)$, for all values of η that are less than one. Using the Tanh activation function, the actor network condenses the activity to the interval (-1,1).

And we start two critic target networks, $Q_1'\left(\mathrm{S}_{t,m}, \mathrm{A}_t' \mid \theta^{Q'}_{m,1}\right)$ and $Q_2'\left(\mathrm{S}_{t,m}, \mathrm{A}_t' \mid \theta^{Q'}_{m,2}\right)$, and one goal actor system, $\mu'\left(\mathrm{S}_{t,m} \mid \theta^{\mu'}_{m}\right)$, where $\mathrm{A}_t' = (\mathrm{a}_{t,1}', \mathrm{a}_{t,2}')$. The factors $\theta^{Q'}_{m,1}, \theta^{Q'}_{m,2}$ and $\theta^{\mu'}_{m}$ use the initialization values of the existing actor networks that correspond to them. The course of action $\mathrm{a}_{t,m}'$ is specified as:

$$\mathrm{a}_{i,m}' = \mu'\left(\mathrm{S}_{i+1,m} \mid \theta^{\mu'}_{m}\right) + \epsilon_1 \tag{15}$$

Algorithm: MASAC-RA (CTDE, twin critics)

```
Init for each agent m ∈ {1:SubCh, 2:Power}:
  policy πθm (Gaussian→Tanh), critics Qϕm,1, Qϕm,2, targets ϕ̄m,i←ϕm,i
  temperature αm (log-param ζm); replay D=Ø
for episodes do
  s ← reset()
  repeat
    // decentralized execution
    for m in {1,2}: a_m ~ πθm(·|s); logπ_m ← log πθm(a_m|s)
    a ← [a_1,a_2];   a ← enforce_constraints(a)  // power, RB budgets, slice isolation
    (r, s', done) ← step(a);  push (s,a,r,s',done) to D;  s ← s'
    // learning (periodically)
    if update_step:
      sample B tuples from D
      // next joint action
      for m: a'_m ~ πθm(·|s'); ℓ'_m ← log πθm(a'_m|s');   a' ← [a'_1,a'_2]
      // targets and critic updates
      for m: y_m ← r + γ(1−done)[ min_i Qϕ̄m,i(s',a') − αm ℓ'_m ]
          ϕm,i ← AdamStep ∇(Qϕm,i(s,a) − y_m)^2
      // actor updates (per agent; others fixed to current policies)
      for m: â_m ~ πθm(·|s); â ← replace a with â_m
          θm ← AdamStep ∇[ αm log πθm(â_m|s) − Qϕm,1(s,â) ]
      // temperature and soft targets
      for m: ζm ← AdamStep ∇[ −αm (log πθm(â_m|s) + H*) ]; αm←e^{ζm}
      for m,i: ϕ̄m,i ← τϕm,i + (1−τ)ϕ̄m,i
  until done
end for
```

3.5. Reward

We imagine Changes to the environment occur when the two agents' actions $\mathrm{a}_{(t,m)}$ are carried out, from $\mathrm{S}_{t,m}$ to $\mathrm{S}_{t+1,m}$. With *m*=1,2, the environment provides the m-th agent with a reward $r_{(t,\ m)}$. To achieve the identical goal in upper-level optimizing, two agents are tasked with allocating sub-channels and power resources. Hence, at time *t*, we assign the identical reward function to both agents, which is, $r_{t,1} = r_{t,2}$, in accordance with the upper-level optimization's objective function and the violation of constraints *eq. (1)*.

$$r_{t,m} = \sum_n c_n \sum_k v_{n,k} R_{n,k} + \lambda \sum_n \sum_{u \in u_n} b_{n,u}(t) - \varrho\varrho,\ m = 1,2 \tag{16}$$

In which the level of constraint violation is denoted by ϱ and the punishment coefficient is represented by *t*. Hence, the overall benefit $R^{\text{total}}_{t,m}$ the *m*th agent may be expressed as

$$R^{\text{total}}_{t,m} = \sum_{\tau=0}^{T} \gamma^{\tau} r_{t+\tau,m} \tag{17}$$

where $\gamma \in [0,1]$ acts as a discount component. A function that relies on the Belman function, known as the Q-value function, may be used to assess the predicted total return per action. One way to represent it is like this:

$$\begin{aligned} Q^{\pi}(S_{t,m}, a_{t,1}, a_{t,2}) &= E_{\pi}\left[R^{\text{total}}_{t} \mid S_{t,m}, a_{t,1}, a_{t,2}\right] \\ &= E_{\pi}\left[\sum_{\tau=0}^{T} \gamma^{\tau} r_{t+\tau,m} \mid S_{t,m}, a_{t,1}, a_{t,2}\right] \\ &= E_{\pi}\left[r_{t,m} + \gamma Q^{\pi}(S_{t+1,m}, a_{t+1,1}, a_{t+1,2}) \mid S_{t,m}, a_{t,1}, a_{t,2}\right] \end{aligned} \tag{18}$$

We select actions $\mathrm{A}_t = (\mathrm{a}_{t,1}, \mathrm{a}_{t,2})$ For a given state S_t, the agents are defined by *equation (18)*. We proceed by taking action $\mathrm{a}_{t,m}$ to get the rewards of agents $r_t = (r_{t,1}, r_{t,2})$ and the novel states of the two agents $\mathrm{S}_{t+1} = (\mathrm{S}_{t+1,1}, \mathrm{S}_{t+1,2})$. Transition ($\mathrm{S}_t, \mathrm{A}_t, r_t, \mathrm{S}_{t+1}$) data is saved in memory replay *D*, as seen in Algorithm 1's line 13.

We extract samples ($\mathrm{S}_i, \mathrm{A}_i, r_i, \mathrm{S}_{i+1}$) starting from *D* and training the networks at every step with batches of size *N*. Agent 1's training procedure in SAC is shown in *figure 3*. A reduction in loss is achieved by adjusting the settings of the existing critic networks. Here is the loss function for the *m*th agent's *j*th current critic network:

$$L\left(\theta_{m,d}^{Q}\right)=\frac{1}{N}\sum_{i}\left[y_{l,m}-Q_{l}\left(\mathrm{S}_{i,m},\mathrm{A}_{i}\mid\theta_{m,d}^{Q}\right)\right]^{2},j=1,2 \quad (19)$$

where $y_{i,m}=r_{i,m}+\gamma Q'_{target}$ serves as a rough estimate of policy. The desired critic networks $Q1'$ and $Q2'$ have minimal Q-values, which are represented by the values of Q'_{target}'. In other words,

$$Q'_{\text{target}}=\min\left(Q'_{1}\left(\mathrm{s}_{i+1,m},\mathrm{A}'_{i}\mid\theta_{m,1}^{Q'}\right),Q'_{2}\left(\mathrm{s}_{l+1,m},\mathrm{A}'_{i}\mid\theta_{m,2}^{Q}\right)\right) \quad (20)$$

where $\mathrm{A}'_{i}=\left(\mathrm{a}'_{i,1},\mathrm{a}'_{i,2}\right)$ both agents' target actor networks' activities are contained inside. After that, the settings $\theta_{m,j}^{Q}$ it updates the *m*th agent's current critic network to reduce the loss function for the *j*th agent. In other words,

$$\theta_{m,j}^{Q}\leftarrow\arg\ \min L\left(\theta_{m,j}^{Q}\right),j=1,2 \quad (21)$$

A stochastic policy gradient technique is used to update the settings of the agents' current actor networks. We select the *Q*-value from the first present critic network in this study, but any of current critic networks may have yielded the same result. So, for the first agent (m=1), we may calculate the ensemble objective gradient in the following way:

$$\nabla_{\theta_{1}^{u}}J=E\left[\nabla_{\mathrm{a}}Q_{1}\left(\mathrm{S}_{t,1},\mathrm{a},\mathrm{a}_{t,2}\mid\theta_{1}^{Q}\right)\nabla_{\theta_{1}^{\mu}}\mu\left(\mathrm{S}_{t,1}\mid\theta_{1}^{\mu}\right)\Big|_{\mathrm{a}=\mu\left(\mathrm{s}_{t,1}||_{1}^{\mu}\right)}\right] \quad (22)$$

For the second agent,

$$\nabla_{e_{2}^{u}}J=E\left[\nabla_{\mathrm{a}}Q_{1}\left(\mathrm{S}_{i,2},\mathrm{a}_{t,1},\mathrm{a}\mid\theta_{1}^{Q}\right)\nabla_{\theta_{2}^{\mu}}\mu\left(\mathrm{S}_{t,2}\mid\theta_{2}^{\mu}\right)\Big|_{\mathrm{a}=\mu\left(\mathrm{s}_{t,2}|_{2}^{\mu}\right)}\right] \quad (23)$$

Here, we use the Adam optimizer using a learning rate of $\alpha=0.001$ and $\beta_{1}=0.9,\beta_{2}=0.999$ so that the existing actor networks' parameters may be updated. During the training phase, the learning rate α may be changed.

The following is how the parameters of the *m*th agent's target critic networks are updated after a training epoch:

$$\theta_{m,j}^{Q'}\leftarrow\varsigma\theta_{m,j}^{Q}+(1-\varsigma)\theta_{m,j'}^{Q'j=1,2} \quad (24)$$

Every *m*th agent's target actor network has its parameters updated in the following way:

$$\theta_{m}^{l'}\leftarrow\varsigma\theta_{m}^{\mu}+(1-\varsigma)\theta_{m}^{l'} \quad (25)$$

where $\varsigma<1$ is an updated target network that uses a reduced constant. An actor's network takes the state St as input, applies a policy depending on the action, and then returns the *Q*-value. The policy enhancement is predicated on the *Q*-value. The actor's estimation of *Q*-values *via* learning of temporal differences is the task of the critic network when evaluating policies.

$$\mathrm{L}=\mathrm{F}\left[\left(\mathrm{R}_{\mathrm{t}}+\gamma\mathrm{Q}\left(\mathrm{St}+1,\mu_{\emptyset}(\mathrm{St}+1)\right)-\mathrm{Q}(\mathrm{St},\mathrm{Ac})\right)^{2}\right] \quad (26)$$

Where, $[(\mathrm{R}_{\mathrm{t}}+\gamma\mathrm{Q}\left(\mathrm{St}+1,\mu_{\emptyset}(\mathrm{St}+1)\right)$ depict the ideal *Q*-value at time *t*. For larger projected *Q*-values, the critic network determines the direction of action change by calculating the gradients$\nabla_{a}Q(St,Ac)$. The theorem of stochastic policy gradient is used to calculate the gradient performance $\nabla_{\emptyset}I(\mu_{\emptyset})$, which is used to assess the actor weight value.

$$\nabla_{\emptyset}I(\mu_{\emptyset})=E_{s\sim\sigma^{\mu}}\left[\nabla_{a}Q(St,Ac)|_{a=\mu_{\emptyset(St)}}\ \nabla_{\emptyset}\mu_{\emptyset}(St)\right] \quad (27)$$

An essential function in reinforcement learning is the upkeep of efficient improved exploitation exploration.

Both the current *Q*-value Q(S$_t$, A$_c$) while the ideal *Q*-value is estimated from *equation (27)* using the online network $\mathrm{Q}[(\mathrm{R}_{\mathrm{t}}+\gamma\mathrm{Q}\left(\mathrm{St}+1,\mu_{\emptyset}(\mathrm{St}+1)\right)$. After then, by keeping tabs on the weight values of the online networks, the ideal network's weights are adjusted. Instead of learning only one *Q*-value, the suggested TD2PG method learns the environment through concurrently learning two *Q*-functions. The *Q*-value was learned utilizing QL and DQL algorithms by a double critic network, which was applied in this case. The following is a definition of the *Q*-value computation for DQL and QL,

$$x^{QL}=R_{t}+\gamma Q(St+1,arg\ \underset{Ac+1}{\mathrm{Max}}\,Q(St+1,Ac+1)) \quad (28)$$

$$x^{DQN}=R_{t}+\gamma Q(St+1,arg\ \underset{Ac+1}{\mathrm{Max}}\,Q(St+1,Ac+1)) \quad (29)$$

In this case, the QL evaluated the action using the same *Q*-table. When evaluating the action, DQN also utilized the same weight value as the neural network. The policy $\mu\emptyset$ in TD2PG is fine-tuned in relation to the critic's value *Q*. However, the target updates the *Q*-value using a similar metric, which might lead to an overestimation of *Q* and impact the policy's quality.

Our suggested double QL technique calculates the *Q*-values for both the actor and the critic, addressing the issue of single estimation of *Q*-values and limiting the danger of overstated *Q* values. We present two networks, *Q1* and *Q2*, to provide that function. The following is a definition of the outcome of estimating the *Q* values of two networks,

$$x=R_{\tau}+\gamma\min_{i=1,2}Q_{i}\left(St_{\tau+1},\mu_{clip}(St_{\tau+1})\right) \quad (31)$$

The following is a definition of the training data used to assess the error value of TD,

$$L_{i}=M^{-1}\sum_{m-1}^{M}\delta_{m,i}^{2}$$

The following is a definition of the TD error for each critic network, where *M* stands for the experiences,

$$\delta_{m,i}=R_{m,i}+\gamma\underset{i=1,2}{\mathrm{Min}}\,Q_{i}(St_{m+1,i}\,\mu_{clip}(St_{m+1,i}))-Q_{i}(St_{m,i}Ac_{m,i}) \quad (32)$$

The following is a definition of the policy gradient that the actor updates:

$$\nabla_{\emptyset} I(\mu_{\emptyset}) \quad = M^{-1} \sum_{m=1}^{M} \nabla_a Q_1(St_{m,1}, a)|_{a=\mu_{\emptyset}(St_{m,1})} \ \nabla_{\emptyset}\mu_{\emptyset}(St_{m,1}) \tag{33}$$

Finally, the critic network is fed the weight values.

$$w_i \ \leftarrow w_i - \propto^{w} \nabla_i L_i \tag{34}$$

Following the learning process, the following updates are made to the actor along with target network,

$$\emptyset \leftarrow \emptyset - \propto^{\emptyset} \ \nabla_{\emptyset} I(\mu_{\emptyset})$$

$$w_i' \leftarrow v w_i + (1-v) w_i' \ \ and\ \emptyset' \leftarrow v\emptyset + (1-v)\emptyset' \tag{35}$$

Where, $\propto^{w}$ and $\propto^{\emptyset}$ stand for the rate of soft update and denote the learning rate variable for the gradient descent technique. The suggested TD2PG determines the state of the environment using the learning rate.

In order to guarantee complete reproducibility of MASAC-RA, we give detailed information about implementation, which includes the architecture of the neural networks, all the training hyperparameters, and experimental conditions. All actors have two 128-unit ReLU layers and all centralized twin-critic networks have two 256-unit ReLU layers. The training is configured with 2400 episodes with 150 steps per episode and 1M transitions replay buffer, 256 as batch size, and five random seeds (04). Both networks are trained using Adam (initial learning rate of 3×10 $^{-4}$, 0.005, 0.99) and SAC temperature 8 is learned with an update rate of 1x10^{-4}. All parameters are summarized in a special hyperparameter table, and a lightweight code release, such as environment, agent, replay buffer, and plotting scripts, are provided to ensure the independent check of the results.

4. RESULTS

Here we detail the results of the multi-criteria mathematical evaluation of the 5G network's DRL resource distribution system.

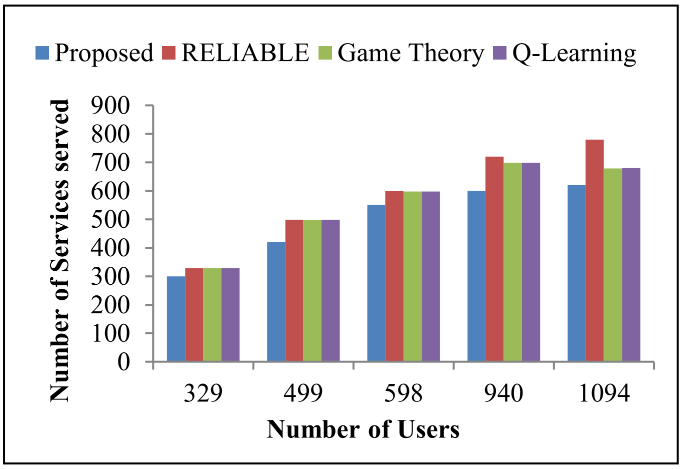

Figure 3. Number of Services served

We used Python to implement DRL's technique. Because the Random Waypoint Mobility system's stop time allows users to remain at a position in the city (like a convenience shop) for a long, we imagined that DRL would be implemented in an urban environment made of individuals moving following this model. *figure 3* shows the Number of users and services served for the connectivity of MEC devices. We assumed a Pearson Type III distribution for the user's input and output as well as MEC for the DRL assessment.

In order to illustrate various scenarios and compare the best and worst-case scenarios, the simulation took into account a range of user numbers (*i.e.*, 327, 499, 596, 930, among 1088) to represent various scenarios. We used a Manhattan neighborhood where four linked 5G cell towers provided coverage for the urban environment and enabled connectivity of MEC devices to illustrate the urban situation, which is depicted in *Table 1*.

TABLE 1. Simulation Setup

Parameter	**Value**
Maps	Manhattan city
Number of users	327, 499, 596, 930 and 1088
Cellular Network	Four connected 5G cell towers
User input and output in MEC	Pearson Type III distribution
Services	Security and entertainment
Security service time	1 h
Security service bandwidth consumption	1%
Security service memory consumption	0.5%
Security service processing consumption	1.5%
Entertainment service time	2 h
Entertainment service bandwidth consumption	4%
Entertainment service memory consumption	2.5%
Entertainment service processing consumption	2.5%

In light of the findings in [18], we gave each MEC device the option of two service kinds. In particular, the following factors were considered while deciding which security service to prioritize: *(1)* one hour of service implementation time; *(2)* one percent of 5G network bandwidth usage; *(3)* five percent to process consumption; and *(4)* one and a half percent of memory consumption. In contrast, the following were features of the second service the entertainment service: *(i)* a service execution duration of 2 hours; *(ii)* a bandwidth consumption of 4% when taking 5G communication into account; *(iii)* a processing consumption of 2.5%; and *(iv)* a memory consumption of 2.5%.

Thus, we verified the effect of distributing resources from various categories of services by evaluating these services throughout three scenarios. Situation 1 depicts a request for only security services, Situation 2 shows a request for only entertainment services, and Situation 3 depicts a random choice between requesting both services and which one will be requested.

To evaluate DRL's efficacy in this context, we used three different methods of allocating resources. However, Best sorts the available resource numbers into a list, passes through the controller, and then selects the most resource-intensive MEC gadget based on this list. Last but not least, Worst is quite identical to the previous one; it also goes through the controller, but this time it calculates the numbers of available resources, sorts them into a list, and then selects the best MEC device to try the service. Furthermore, we evaluate our approach in comparison to two other paradigms that use the Best and Worst cases for memory allocation. This would not be an accurate comparison with other approaches as the current allocation algorithms did not take into account the anticipated level of mobility with each service's duration.

When evaluating various methods of allocating resources, we take into account the following metrics:

- The number of services provided is the total number of services allocated in a MEC device.
- The number of services blocked is the total number of wrong service allocation decisions made because there weren't enough resources to go around. For this reason, the service will remain unavailable until DRL discovers a MEC device capable of allocating.

The total number of offerings denied represents the total number of needs that were not allocated through any MEC because of insufficient resources. *Figure 4* shows that DRL also has a lower blocking time for switching across MEC devices, regardless of the number of requests, low or large. All of the other approaches behave similarly since they all allow for more service mobility across the MEC devices.

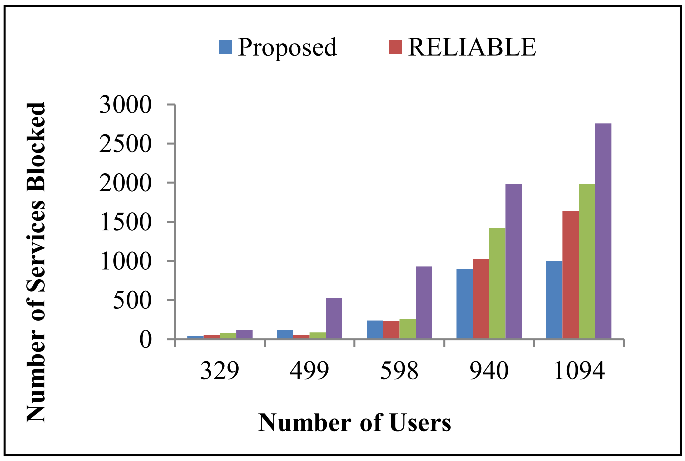

Figure 4. Number of Services Blocked

See *figure 5* for an illustration of how the system's service use is significantly impacted by insufficient resource balance. Due to the fact that their allocation strategies failed to perform load balancing across the various MECs, the three methods Greedy, Best, and Worst exhibited identical behavior, causing a subset of MECs to become overloaded. Consequently, DRL was able to decrease the total amount of services refused by 20% as a result of an effective approach for balancing the flow of information.

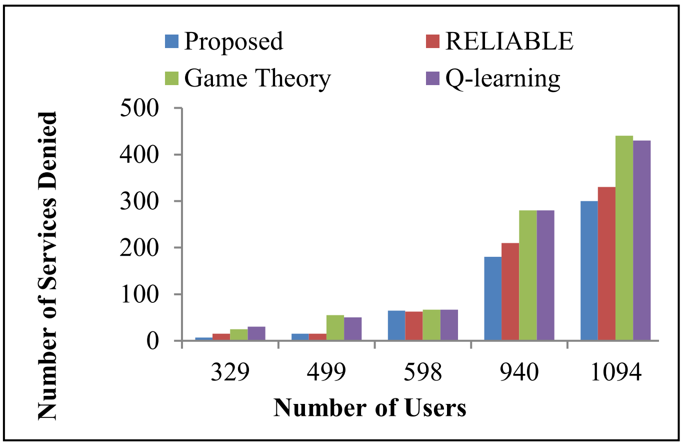

Figure 5. Number of Services Denied

Scenario 2's findings show that DRL decreases block count while increasing request throughput; this is in contrast to taking into account requests for purely entertainment services, which necessitate more processing power, bandwidth, and time for service execution. When we look at Scenario 2 in comparison to Scenario 1, we can see that DRL keeps the total number of services provided high and the number of refused services low when the service requests more resources. This is due to the fact that DRL takes the flow's mobility and execution time into account in its decision policy, independent of the characteristics pertaining to the processing capability of the MEC device. So, when we compared the outcomes of Scenario 1 and Scenario 2, we found that both scenarios led to a rise in users. Although DRL exhibited behavior comparable to the other solution in Scenario 1 using fewer users (329, 499, 595), it outperformed it in the transaction and scenario using a larger number of users (925, 1109) by balancing its computational load through mobility prediction, impact allocation, or changes in flow. Scenario 2 shows the same thing. However, in Situation 2, it also worked effectively with fewer users since the resources that these individuals made accessible were larger, allowing DRL to better manage the resources that were available.

In all three conditions (Security-only, Entertainment-only, Mixed) the numbers of Services Served with MASAC-RA are mostly greater than in case of Blocked and Denied. This benefit is more evident with increased user densities (930 and 1088 users) when they are at higher load. This tendency indicates that the learned policy can take advantage of long-term organization in the environment, *i.e.* the patterns of mobility, the time spent on the service, or the distributions of MEC loads, and not resort to instant availability of resources.

There are also converging episode-return curves that indicate early oscillations due to exploration of policy by entropy regularization, and late plateaus which indicate that agent have learned to avoid violating constraints (*e.g.*, in overloaded MEC nodes, RB/power constraints) and to maximize throughput-reward functions. The fact that the parameter of the decreasing temperature proves that, as the training goes on, the learned policy becomes more deterministic.

Heuristic baselines stabilize fast as they do not look down the line: they greedily assign to the first or optimal MEC node without foresight of shortages of downstream capacity. The result is that it causes groupings of overloaded MEC servers, high blocking rates and variable service quality, which MASAC-RA eliminates through learning a mobility- and load-aware allocation pattern.

It is evident that MASAC-RA performs significantly better than any of the heuristic baselines in all user density conditions as seen in *table 2*. The method has better served services and low numbers of the blocked and denied services particularly in high load conditions. In contrast to Greedy or Best-fit, which bases their short-term decisions on the present availability only, MASAC-RA learns about long-term allocation patterns by means of its centralized critics, which allows it to make better load balancing decisions and reduce violation of constraints. This means that MASAC-RA offers significantly better throughput and more reliable performance, and it is seen to have definite benefits in dynamic and congested MEC settings.

TABLE 2. Overall Performance (Mean ± Std)

Method	327 Users	499 Users	596 Users	930 Users	1088 Users
Services Served (↑)					
MASAC-RA (ours)	94.8± 0.6	93.2 ± 0.8	91.5 ± 1.0	88.7 ± 1.1	86.9 ± 1.3
Greedy	87.9± 0.8	84.6 ± 1.0	82.4 ± 1.2	78.1 ± 1.5	75.6 ± 1.6
Best-fit	89.3± 0.7	86.0 ± 0.9	83.1 ± 1.1	79.5 ± 1.4	77.1 ± 1.6
Worst-fit	72.4± 1.4	70.3 ± 1.6	68.5 ± 1.8	66.0 ± 2.0	63.7 ± 2.1

5. CONCLUSIONS

In this paper, we provide a strategy that uses MEC to solve the issue of 5G network resource allocation. We take into account a MEC network made up of a collection of mobile devices that may pool their resources to provide more services. We achieved this by developing a multi-criteria decision-making approach, whereby SAC is one of many factors taken into account, including service and network characteristics as well as flow mobility. Consequently, the decision-making process maximizes Cloud resource use by providing a balanced input having varying degrees of relevance. Numerical findings demonstrate that the suggested approach reduces the total number of service blocks and the number of services refused by balancing the distribution of resources, allowing for a higher quantity of services to be offered. To further enhance the process, we will take into account more aspects in future studies, including mobility and energy use. The applicability and effectiveness of the suggested algorithm were proven by the outcomes. In the future, we want to investigate ways to realistically train a single agent per slice by allocating varying quantities of resource blocks and fixed input action dimensions. A lot of limitations in the real landing are brought about by the fact that this form of neural network-based reinforcement learning uses various dimensional resource allocation during training. In order to increase the algorithm's generalizability, we will implement some changes in this area.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1] O. Abuajwa, M. B. Roslee, Z. B. Yusoff, L. L. Chuan, and P. W. Leong, “Resource Allocation for Throughput versus Fairness Trade-Offs under User Data Rate Fairness in NOMA Systems in 5G Networks,” *Appl. Sci.*, vol. 12, no. 1, pp. 1–20, 2022.

[2] S. S. Sefati *et al*., "A Comprehensive Survey on Resource Management in 6G Network Based on Internet of Things," in *IEEE Access*, vol. 12, pp. 113741-113784, 2024.

[3] A. Minalkar, S. Doss, and R. Doshi, “A Study of the Resource Allocation Mechanism for Secure Video Transmission in 5G Networks,” in *Proc. 2023 IEEE Int. Conf. on Contemporary Computing and Communications (InC4)*, 2023, pp. 1–6.

[4] Y. L. Lee and D. Qin, "A Survey on Applications of Deep Reinforcement Learning in Resource Management for 5G Heterogeneous Networks," *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Lanzhou, China, 2019, pp. 1856-1862.

[5] B. Agarwal, M. A. Togou, M. Ruffini, and G. Muntean, “A Low Complexity ML-Assisted Multi-Knapsack-Based Approach for User Association and Resource Allocation in 5G HetNets,” in *Proc. IEEE Int. Symp. on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2023, pp. 1–6.

[6] F. Bahramisirat, M. A. Gregory and S. Li, "Multi-Access Edge Computing Resource Slice Allocation: A Review," in *IEEE Access*, vol. 12, pp. 188572-188589, 2024.

[7] P. K. Rebari and B. R. Killi, “Deep Learning Based Traffic Prediction for Resource Allocation in Multi-Tenant Virtualized 5G Networks,” in *Proc. TENCON 2023 – IEEE Region 10 Conf.*, 2023, pp. 97–102.

[8] S. O. Oladejo and O. E. Falowo, "Latency-Aware Dynamic Resource Allocation Scheme for 5G Heterogeneous Network: A Network Slicing-Multitenancy Scenario," *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona, Spain, pp. 1-7, 2019.

[9] F. Allagiotis, C. Bouras, V. Kokkinos, A. Gkamas, and P. Pouyioutas, “Reinforcement Learning Approach for Resource Allocation in 5G HetNets,” in *Proc. 2023 Int. Conf. on Information Networking (ICOIN)*, 2023, pp. 387–392.

[10] S. Kumar, R. Mahapatra and A. Singh, "Power allocation in 5G HetNets: A Federated Learning Approach," *2022 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Gandhinagar, Gujarat, India, pp. 275-280, 2022.

[11] S. Ghosh and D. De, “TARA: Weighted Majority Cooperative Game Theory-Based Task Assignment and Resource Allocation in 5G Heterogeneous Fog Network for IoT,” *J. Supercomput.*, vol. 79, pp. 14633–14683, 2023.

[12] K. Tsachrelias, C. Katsigiannis, V. Kokkinos, A. Gkamas, C. Bouras, and P. Pouyioutas, “Optimizing Resource Allocation in 5G MIMO Networks Using DUDe Techniques,” in *Proc. 2024 14th Int. Symp. on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2024, pp. 454–459.

[13] C. Wang and W. -C. Hsiao, "Resource Allocation using Artificial Intelligence for Vehicle-to-Everything (V2X) Communications on Licensed and Unlicensed Spectrum," *2024 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Kaohsiung, Taiwan, pp. 1-4, 2024.

[14] N. M. Laboni et al., “A Hyper Heuristic Algorithm for Efficient Resource Allocation in 5G Mobile Edge Clouds,” *IEEE Trans. Mobile Comput.*, vol. 23, pp. 29–41, 2024.

[15] M. Talaat Fahim, M. Ibrahim, and N. M. Elshennawy, “Efficient Resource Allocation of Latency Aware Slices for 5G Networks,” *J. Eng. Res.*, 2023.

[16] J. Xu, “Efficient Trajectory Optimization and Resource Allocation in UAV 5G Networks Using Dueling-Deep-Q-Networks,” *Wireless Netw.*, pp. 1–11, 2023.

[17] Y. -H. Tu, Y. -W. Ma, Z. -X. Li, J. -L. Chen and K. Tsukamoto, "Applying Deep Reinforcement Learning for Self-organizing Network Architecture," *2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII)*, Sapporo, Japan, pp. 16-20, 2023.

[18] R. Dubey, P. K. Mishra, and S. Pandey, “SGR-MOP Based Secrecy-Enabled Resource Allocation Scheme for 5G Networks,” *J. Netw. Syst. Manag.*, vol. 31, pp. 1–26, 2023.

[19] H. Zhang, S. Xu, S. Zhang and Z. Jiang, "Slicing Framework for Service Level Agreement Guarantee in Heterogeneous Networks—A Deep Reinforcement Learning Approach," in *IEEE Wireless Communications Letters*, vol. 11, no. 1, pp. 193-197, Jan. 2022.

[20] T. Verma, A. Raza, S. Shrivastava, A. Kumar, D. P. Kothari and U. D. Dwivedi, "A Novel On-Policy DRL-Based Approach for Resource Allocation in Hybrid RF/VLC Systems," in *IEEE Transactions on Consumer Electronics*, vol. 71, no. 1, pp. 550-560, Feb. 2025.

[21] A. K. Tiwari, P. K. Mishra, S. Pandey, and P. R. Teja, “Resource Allocation and Mode Selection in 5G Networks Based on Energy Efficient Game Theory Approach,” *Int. J. Recent Innov. Trends Comput. Commun.*, 2022.

[22] W. Jing, J. Wang, J. Ren, Z. Lu, H. Shao and X. Wen, "Radio Resource Allocation Optimization for Delay-Sensitive Services Based on Graph Neural Networks and Offline Dataset," in *IEEE Transactions on Vehicular Technology*, vol. 74, no. 3, pp. 4510-4525, March 2025.

[23] J. Lin, P. Chou, and R. Hwang, “Dynamic Resource Allocation for Network Slicing with Multi-Tenants in 5G Two-Tier Networks,” *Sensors*, vol. 23, 2023.

[24] S. Lavanya, N. M. S. Kumar, S. Thilagam and S. Sinduja, "Fog computing-based radio access network in 5G wireless communications," *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, pp. 559-563, India, 2017.

[25] H. Tsai, S. Kao, Y. Huang, and F. Chang, “Energy-Aware Mode Selection for D2D Resource Allocation in 5G Networks,” *Electronics*, 2023.

[26] M. Karuppiyan, H. Subramani, S. Kandasamy Raju, and M. Maradi Anthonymuthu Prakasam, “Dynamic Resource Allocation in 5G Networks Using Hybrid RL-CNN Model for Optimized Latency and Quality of Service,” *Netw.*, pp. 1–25, 2024.

[27] J. Menard, A. Al-Habashna, G. A. Wainer, and G. Boudreau, “Distributed Resource Allocation in 5G Networks with Multi-Agent Reinforcement Learning,” in *Proc. 2022 Annual Modeling and Simulation Conf. (ANNSIM)*, 2022, pp. 802–813.

[28] L. Liu, X. Yuan, D. Chen, N. Zhang, H. Sun, and A. Taherkordi, “Multi-User Dynamic Computation Offloading and Resource Allocation in 5G MEC Heterogeneous Networks with Static and Dynamic Subchannels,” *IEEE Trans. Veh. Technol.*, vol. 72, pp. 14924–14938, 2023.

[29] J. Logeshwaran, N. Shanmugasundaram, and J. Lloret, “Energy-Efficient Resource Allocation Model for Device-to-Device Communication in 5G Wireless Personal Area Networks,” *Int. J. Commun. Syst.*, vol. 36, 2023.

[30] M. Khani, S. Jamali, M. K. Sohrabi, M. M. Sadr, and A. Ghaffari, “Resource Allocation in 5G Cloud-RAN Using Deep Reinforcement Learning Algorithms: A Review,” *Trans. Emerg. Telecommun. Technol.*, vol. 35, 2023.

[31] A. Hegde, R. Song, and A. Festag, “Radio Resource Allocation in 5G-NR V2X: A Multi-Agent Actor-Critic Based Approach,” *IEEE Access*, vol. 11, pp. 87225–87244, 2023.

Author Biography:

Y Meghamala is an Assistant Professor in the Electronics and Communication Engineering (ECE) department at the Institute of Aeronautical Engineering (IARE), Hyderabad, Telangana. Teaches undergraduate and postgraduate engineering courses in ECE. Develops and contributes to ICT-enabled learning modules and video lecture content (check out her playlists on YouTube via IARE’s resources). Supports outcome-based education, research initiatives, and student-oriented innovation programs through multiple committees.

Dr. P. John Paul holds a B.Tech in Electronics and Communication Engineering from Nagarjuna University, a Master’s degree in Digital Systems from Osmania University, and a Ph.D. in Computer Science from HCU, specializing in VLSI architecture for SoC multimedia. With over 35 years of experience as an academician, researcher, and administrator, he has made significant contributions to the field of engineering education. Dr. Paul has taught various subjects in ECE, CSE, EEE, and Instrumentation. He has published 13 books, including widely used textbooks such as Electronic Devices and Circuits, which is employed at NIT Hamirpur, Himachal Pradesh.

Dr. M. Aravind Kumar obtained B. Tech Degree in ECE, M.Tech Degree in VLSI System Design from JNTUH and Ph.D from GITAM University, Visakhapatnam. He has 15 years of Teaching experience. He is a Life member of FIE, ISTE, IETE, SCIEI, UACEE, and IAENG. He is one of the Editorial board and Reviewer board members in five international journals. He was awarded as the BEST PLACEMENT Officer in the year 2007 and 2008 from the Jawaharlal Nehru Knowledge center, Hyderabad. He has organized 5 National level workshops/ FDP. His areas of Research.