International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2022/212561 Volume 9, Issue 3, May – June (2022)

RESEARCH ARTICLE

Deep Q-Learning Network-Based Energy and Network-Aware Optimization Model for Resources in Mobile Cloud Computing

Puneet Sharma
Department of Computer Science and Engineering, Amity University, Uttar Pradesh, India
puneetgrandmaster@gmail.com

T. Sakthivel
Firstsoft Technologies Private Ltd., Chennai, Tamil Nadu, India
sakthi@firstsoftech.com

Deepak Arora
Department of Computer Science and Engineering, Amity University, Uttar Pradesh, India
deepakarorainbox@gmail.com

Received: 15 May 2022 / Revised: 10 June 2022 / Accepted: 14 June 2022 / Published: 28 June 2022

Abstract – Mobile Cloud Computing (MCC) enables computation offloading procedures and has become popular in resolving the resource limitations of mobile devices. To accomplish effective offloading in the mobile cloud, modeling the application execution environment with Quality of Service (QoS) is crucial. Hence, optimization of resource allocation and management plays a major role in ensuring the seamless execution of mobile applications. Recently cloud computing research has adopted the reinforcement learning models to optimize resource allocation and offloading. In addition, several optimization mechanisms have considered the network transmission rate while selecting the network resources. However, mitigating the response time becomes critical among the dynamically varying mobile cloud resources. Thus, this paper proposes a joint resource optimization methodology for the processing and network resources in the integrated mobile-network-cloud environment. The proposed approach presents the Energy and Network-Aware Optimization solution with the assistance of the Deep Q-learning Network (ENAO-DQN). Designing an energy and network-aware resource optimization strategy recognizes the quality factors that preserve the device energy while allocating the resources and executing the compute-intensive mobile applications. With the potential advantage of the Deep Q-learning model in decision-making, the ENAO-DQN approach optimally selects the network resources with the enrichment of the maximized rewards. Initially, the optimization algorithm prefetches the quality factors based on the mobile and application characteristics, wireless network parameters, and cloud resource characteristics. Secondly, it generates the allocation plan for the application-network resource pair based on the prefetched quality factors with the assistance of the enhanced deep reinforcement learning model. Thus, the experimental results demonstrate that the ENAO-DQN model outperforms the baseline mobile execution and cloud offloading models.

Index Terms – Mobile Cloud Computing, Resource Allocation, Optimization, Energy Consumption, QoS, Deep Reinforcement Learning, Q-learning, Wireless Network Resource.

1. INTRODUCTION

Mobile Cloud Computing (MCC) enables the offloading procedure to execute the data-intensive requests generated from the mobile to the remote cloud. MCC paradigm [1] preserves the energy and storage of the mobile device by remotely executing the compute-intensive requests using the cloud resources. MCC assists mobile devices with improved reliability, scalability, processing power, data storage facility, battery lifetime, and dynamic service provisioning. The computation offloading model heavily relies on resource allocation to effectively execute the end-user requests on the remote server with the improved Quality of Service (QoS) [2]. An efficient resource allocation aids in ensuring the QoS while providing cloud services to the mobile users in the distributed environment [3]. The optimization of the resource allocation approach improves the QoS by considering the response time, profit, energy consumption, and latency parameters in the mobile cloud environment. In the real world, most scientific mobile applications utilize the cloud resources such as social networks, gaming, finance, linguistics, economics, engineering, geophysics, and mathematical fields. The optimized resource allocation achieves the mobile user anticipations of QoS requirements without SLA violations in the mobile cloud environment [4].

RESEARCH ARTICLE

Computation offloading encounters numerous issues during the transmission over the increased user mobility, instability, uncertainty, and fluctuation of the network resources. Most of the existing models [5, 6] lack to consider the communication level optimization regardless of analyzing the resource constraints in the mobile device, application-level preferences, and available network resources instead of performing the computation level optimization of resources in the cloud. Several resource allocation models have presented the network resource selection model to overcome this constraint to ensure execution optimization in the MCC environment. However, the existing optimization models [7, 8] attempted to tackle the provisioning of the energy-aware seamless execution in the dynamic mobile cloud. Several existing research works [9, 10] have developed machine and deep learning models for MCC to handle offloading and resource optimization constraints. Among different learning models [11], the reinforcement learning-based approach [12] has gained significant attention in the sequential offloading decision-making. The reinforcement learning model considers the future reward for the environment and adjusts the allocation or decision-making policy through the agent, which is an effective solution for the time-variant systems. In particular, the deep neural network approach with Q-learning effectively handles the large-scale data in the dynamic environment [13]. Hence, to ensure resource allocation optimization, this work focuses on the allocation of the network as well as processing resources with the assistance of the deep learning model to preserve the mobile Device Energy (DE) during the execution of compute-intensive applications. The proposed Energy and Network-Aware Optimization approach with the Deep Q-learning Network (ENAO-DQN) model employed the Deep Reinforcement Learning (DRL) model for allocating network resources with optimally improved response time and reduced DE. The ENAO-DQN model presents the resource optimization algorithm to select the wireless network resources based on the quality factors. The major contributions are presented below.

- This research presents an Energy and Network-Aware resource Optimization (ENAO-DQN) model based on the Deep Q-learning (DQN) to reduce the DE through the network resource selection.
- By modeling the energy and network-aware resource selection algorithm, it enriches the wireless network interface selection by adopting the enhanced DQN-based decision-making from the knowledge of the multiple reinforcement learning agents.
- Thus, the proposed approach allocates the mobility-aware resources considering energy and network resources to support the local and global rewards for the quality factors, which improves the application response time under heterogeneous wireless networks.

The rest of the paper is organized as follows: Section2 discusses the related works of recent mobile cloud offloading and optimization methods of resource allocation. The problem statement, the system model of the MCC environment, and the reinforcement learning model are provided in Section 3. The proposed optimization strategy for the mobile cloud resource allocation is described in section 4. The details of the implementation and performance evaluation of the proposed optimization model are illustrated in section 5. Finally, section 6 concludes the paper.

2. RELATED WORKS

This section reviews several existing research contributions in the computation offloading and resource allocation approaches for the MCC environment and edge computing. Moreover, surveys of conventional offloading and resource allocation methods show that machine learning and deep learning models are paramount for solving these problems.

2.1. Resource Allocation and Computation Offloading Approaches

The energy-optimal framework [14] preserves the DE during the transmission, and the application is offloaded from the mobile. An energy model is proposed by applying the Lyapunov optimization to decrease the energy utilization in the device during the application transmission between the mobile and the remote cloud [15]. Although, stochastic wireless channel optimization-based energy minimization degrades its performance during the computation of intensive applications and multi-task offloading. The resource allocation optimization model Q-MAC [16] takes account of QoS and mobility factors while allocating the resources for effectively offloading mobile application tasks to the cloud and cloudlets. It ensures the time efficiency of the application execution even when there is increased mobility among the users. However, it is limited to delay-tolerant applications and fails to contemplate the scheduling order in the execution environment due to the joint scheduling-offloading strategy. The joint offloading and resource allocation model [17] presents a three-step optimization algorithm that consists of the Semi Definite Relaxation (SDR) approach, Alternating Optimization (AO) approach, and Sequential Tuning (ST). Consequently, joint offloading optimization, allocation of the network, and computing resources enforce the mitigation of the computation cost, delay, and energy cost. Despite, lack of the contemplation of QoS factors in the dynamic environment leads to ineffective decision-making in the massive set of resources. The elastic resource provisioning model [18] preserves the trade-off between the DE and resource utilization to improve mobile user experience in the hybrid mobile cloud. By elastically modeling the resource provisioning for the integrated infrastructures of the local and public cloud, it effectively supports the execution of the offloaded tasks or applications. However, ineffective selection

RESEARCH ARTICLE

of network resources negatively impacts the overall energy consumption and the response time instead of the elastic computation resource provisioning. The energy-sensitive cross-layer resource allocation [19] enables the execution of compute-intensive tasks to the distant cloud and the execution of the delay-sensitive computations on the local cloud. It optimizes the system performance of the cloud by regulating the collaboration and cooperation among the public cloud supplier, local cloud agent, and mobile cloud user. Although, the lack of optimizing the communication resources significantly affects the energy consumption and degrades the performance of the delay-sensitive applications. Joint Computation Offloading and Resource Allocation Optimization (RAO) (JCORAO) model is employed in [20] to minimize the offloading latency through the computation offloading and uplink power allocation, uplink subchannel allocation, and resource scheduling. However, realizing the factors that optimally minimize the execution and communication time and save the energy of the mobile device becomes challenging. The research work [21] develops a Multi-User and Multi-Task Offloading (MUMTO) model using divisible Semi-Definite Relaxation (SDR) for jointly computing the offloading decision as well as the allocation of network resources. It models the MUMTO-C algorithm, incorporating the generalized MUMTO SDR with Computing Access Point (CAP) and alternating optimization methods with sequential tuning. The lack of considering the workload requirements and delay limits of different tasks on different computation resources is likely to increase the energy consumption during multi-task offloading. The adaptive two-level resource allocation model [22] initially focuses on the QoS metrics and constraints in the system at the stage of the edge server and linear controllers modeling using a set of linear systems. Subsequently, it maximizes the number of offloaded tasks by optimally balancing the load and application placement in the mobile cloud. Even though it allocates the resources based on the on-demand resource estimation, the lack of considering the network resources affects the overall performance of the application execution. The dynamic resource allocation approach [23] schedules the data-intensive mobile applications on the integrated mobile cloud. It considers the input data size, application complexity, DE, and network bandwidth during the resource allocation. However, the lack of optimally exploring the communication loss and intelligently selecting the resources leads to higher response time. The energy eFFicient framewORk for offload communicaTion (EFFORT) approach [24] targets the communication offloading in the MCC to address the energy consumption constraint in the communication-intensive mobile applications. Even though it considers the battery and the network consumption during the offloading, it lacks to investigate the dynamic resource availability and the iterative selection of optimal resources, which tends to increase response time. A dynamic decision-making-based task scheduling model [25] resolves the execution time and energy consumption constraints through energy-efficient scheduling of the tasks. The task scheduling decision algorithm enhances the decision-making probability of the task processing within minimal time and reduces the power consumption of the mobile device. However, the lack of adopting the intelligence algorithms for predicting the task scheduling on the mobile execution and cloud execution environment misleads the inaccurate decision-making on the massive collection of resources.

2.2. Machine Learning-Based Resource Allocation and Computation Offloading Approaches

To optimize the network and process delay, the Hybrid Delay-aware Workload Assignment (HDWA) [26] efficiently executes the delay-sensitive applications on the cloud server with minimal response time. By applying the reinforcement learning algorithm, it optimally allocates the transmission resources in cloudlet with the minimal transmission delay to effectively offload the computational tasks to the remote server. However, the lack of cooperative offloading and scheduling between the communication and computation resources results in the offloading cost becoming greater than the computation cost due to higher transmission delay. The multi-user mobile edge computing model [27] performs the computation offloading with the delay and energy cost objectives. It jointly optimizes the offloading decision and the resource allocation in the wireless mobile edge computing system with the assistance of the Q-learning-based reinforcement learning model. Although, it fails to minimize the long-term delay cost in the computation offloading over the dynamic network conditions. The adaptive service management model [28] applies the supervised and reinforcement learning strategies in the agent-based mobile cloud computing architecture. Examining the optimal execution location in the mobile cloud environment through the agents minimizes the execution time of the tasks in the multimedia file conversion. However, the agent-based model fails to contemplate the network resources and minimize the communication cost in the mobile cloud environment. Energy-Efficient Deep Learning-based Offloading Scheme (EEDOS) [29] considers the different factors such as the energy available in the user device, energy required for the network conditions, application components, data transfer, computational load, and communication delay while selecting the application components for the execution. By applying the deep learning model, it learns the changing the device and network conditions and generates the optimal offloading decision for the mobile edge computing environment. Even though it addresses a high-dimensional state and action decision constraint, modeling an optimal service composition scheme for the low-power device energy is a key challenge. The deep Reinforcement Learning based Resource Allocation (DRLRA) model [30] adaptively allocates the network and

RESEARCH ARTICLE

computational resources and updates the policy in reinforcement learning for the changeable mobile edge environment. It minimizes the computing and routing delays by effectively selecting the resources, reducing the average application service time, and improving the stability in the mutative environment. However, the decision-making model is single-stage static during the arrival of the tasks in the mobile edge environment. The JointDNN model [31] supports the collaborative computation between the mobile and cloud environment during the inference and training process in the deep learning model. By modeling the optimization at layer granularity in the deep neural networks, it improves the efficiency of the energy and the performance in both the mobile device and the cloud environment. However, it fails to optimize resource utilization regardless of the number of devices and the availability of the network resources. The reinforcement-Learning-based State-Action-Reward-State-Action (RL-SARSA) model [32] addresses the resource management constraint in the mobile edge computing environment. The DRL method performs the task offloading and resource allocation considering the computing delay and energy consumption. Even though it provides a quick solution to the decision-making problem, it fails to guarantee the optimality of the solution.

Deep Meta Reinforcement Learning-based Offloading (DMRO) model [33] handles the network failure and slow learning speed in the dynamic environment by combining the multiple parallel deep neural networks with the Q-learning model for the offloading decision. With the potential advantages of deep learning, reinforcement learning, and meta-learning, it quickly and optimally obtains the offloading decision in the IoT environment. However, it lacks to consider the QoS factors of the response time with the impact of the network resources during the offloading decision in the edge-cloud environment. An autonomous computation offloading model [34] addresses the offloading constraints in executing resource-intensive and time-intensive applications. Adopting the deep neural network, linear regression, and Hidden Markov Model handles the high-dimensional offloading decision-making problem. Even though the autonomous offloading model performs the latency and energy consumption-based predictions in the self-configurable network, constrained resources, limited processing capability, and battery supply are the major constraints for the computation offloading model.

3. MODELING THE OPTIMIZATION STRATEGY FOR MOBILE CLOUD RESOURCE ALLOCATION

Among the millions of cloud-based mobile application categories, mobile gaming applications are time-sensitive that require immediate response and high-intensive resources from the on-demand cloud services. This research work executes the proposed approach on the Memory Arithmetic Unit and Interface (MAUI) MCC architecture [35] to conduct the energy-aware offloading. MAUI architecture is essential in preserving the device energy while offloading and executing the compute-intensive tasks in the remote cloud. To avoid offloading all mobile codes to the remote cloud, MAUI separates the application based on the CPU and network costs on the mobile device during the runtime. The MAUI saves the device energy during the offloading process by highlighting the continuous profiling process-associated highly dynamic offloading framework. Figure 1 shows the modeling of the proposed work that utilizes MAUI-based mobile cloud architecture.

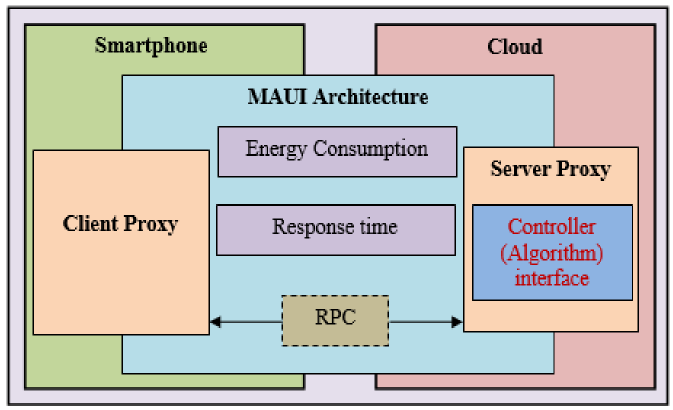

Figure 1 Energy and Network-Aware Resource Selection Algorithm in MAUI Architecture

In the MAUI architecture, the mobile device consists of a solver interface, profiler, and client proxy to select the offloadable tasks of an application based on the requirements and device constraints, especially the energy parameter. The cloud server contains the profiler, solver, server proxy, and controller to execute the offloaded tasks or applications. Especially, the MAUI controller is responsible for executing the offloaded tasks of an application. By utilizing the Remote Procedure Call (RPC), the MAUI architecture establishes the connectivity between the device and the server.

3.1. Problem Statement

MCC enables the execution of the mobile applications on the distant resource-rich cloud server with thin-client connections over the Internet. Hence, high latencies during the application execution become a major constraint in the mobile cloud environment. Accordingly, a lack of network resource management involving the bandwidth selection, latency, and energy efficiency contemplations tends to increase the communication time and reduce user satisfaction. In the MCC, offloading and resource allocation processes must focus on the constraints of the mobile device and QoS to ensure the seamless execution of the cloud-based mobile application in the remote cloud. In addition, lack of optimization of the communication and the communication

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2022/212561 Volume 9, Issue 3, May – June (2022)

RESEARCH ARTICLE

resource selection leads to higher energy consumption and response time among the vast amount of available network and cloud resources, respectively. Hence, it is imperative to contemplate the factors that increase the consumption of energy and response time during the computation offloading and application execution in the MCC.

3.2. System Model

The system model shows the allocation of the resources for the compute-intensive mobile applications in the MCC environment using the reinforcement learning model. In the mobile cloud environment, the number of requests for a particular mobile application is frequently generated from heterogeneous mobile users to which smartphone devices are connected through the Wi-Fi network. Let each mobile device (M_i) has a unique ID for identification, and each mobile user requests similar applications for the execution in the cloud wherein the number of mobile applications i={1, 2,...., x}. In the cloud environment, each data center incorporates a different number of servers with various resource capabilities involving the CPU, memory, and network. The cloud data center is the Cloud Service Provider (CSP) that provides heterogeneous resources to the different application requests. In the cloud data center, R_j refers to a set of cloud resources, j={1,2,...y}.

In the mobile cloud environment, every mobile application execution depends on the network resource selection based on the request characteristics and current network traffic. In the mobile cloud environment, N_k represents the network resources, k={1,2,...z}. In the cloud environment, computation resources involve resource availability, computational storage, power, and network bandwidth. In the mobile cloud environment, the DE heavily relies on executing the i^{th} application on the device, the cloud itself, and the transmission through the Radio Frequency (RF) components. The energy consumption of the RF components is based on energy consumption while sending and receiving the requests between the smartphone and the cloud. As a result, modeling the optimal resource allocation considers two major constraints, the DE and QoS, for executing compute-intensive mobile applications. It heavily relies on the mobile and application characteristics while selecting the network and cloud resources.

3.3. Reinforcement Learning Model

The proposed resource allocation model employs the DRL model to make the allocation decisions. According to the procedure of the DRL model, the agent handles the resource optimization problem. By assessing the decision with the consideration of different criteria, the agents resolve the problem in the resource allocation decision-making. In the proposed model, the criteria considered by the agents are device energy, response time with the inclusion of the communication time, and computation time in the cloud. The proposed resource optimization model is based on the Deep Q-Learning algorithm that belongs to the category of the reinforcement learning algorithm. DRL algorithm targets maximizing the rewards while offloading the tasks and allocating the communication and computation resources in the mobile cloud environment.

- State $\triangleq \{MD_{c(i)}, NR_{c(j)}, CR_{c(k)}\}$ (1)
- Action $\triangleq \{0, 1\}$ (2)
- Reward $\triangleq \alpha \times r_c + \left[\left(\left(\frac{1-\alpha}{2}\right) \times r_{RT}\right) + \left(\left(\frac{1-\alpha}{2}\right) \times r_E\right)\right]$ (3)

In the modeling of reinforcement learning, the research work considers the states of mobile device capability ($MD_{c(i)}$), network resource capability ($NR_{c(j)}$), and cloud resource capability ($CR_{c(k)}$), as mentioned in Equations (1-3). In the DQL-based proposed model, action refers to offloading and resource allocation decision-making, which belongs to either ‘0’ or ‘1’. If the value is ‘1’, the proposed model performs the offloading to the cloud and allocates the network and cloud resources for a particular task with the analysis of the states. Otherwise, the proposed model locally executes the process without allocating the task to that network and cloud resources. The model-free DRL is one of the efficient mathematical models for sequential decision-making in a dynamic MCC environment.

In DQN, the agent calculates the Q-value function, Q(s, a), in which state ‘s’ and action ‘a’ continue until the optimal policy with the successor states is reached. At each time period of decision, the decision-making model iteratively trains the Q-network by adjusting the weights with the target of minimizing the loss function at each time step. In particular, the DQN model adopts the experience replay method as the training method to resolve the non-linear approximations in the Q-network. By randomly selecting the mini-batch of the state-action transitions from the replay memory, the DQN-based decision-making model trains the Q-network.

$$Q(s_t, a) \leftarrow Q(s_t, a) + \alpha\left[r_{t+1} + \gamma \max_{a'_{t+1}} Q(s_{t+1}, a'_{t+1}) - Q(s_t, a)\right] \quad (4)$$

In the formulation of Equation (4), r_{t+1} refers to the reward obtained in the t+1th time slot, and $\gamma \in [0,1]$ refers to the discount factor. If $\gamma = 0$, the current reward is considered by the agent else later rewards. $Q(s_t, a)$ selects the optimal policy for all the actions by $\max_{a'_{t+1}} Q(s_{t+1}, a'_{t+1})$. Moreover, to overcome the curse of dimensionality problem in the storage of state-action pairs in the Q-table, the neural network in the form of DQN. The implementation of the DQN relies on the

RESEARCH ARTICLE

influence of the previous matching relations across the inter-dependence between the user preferences or request characteristics and resources.

The calculation of the energy (E) and response time (RT) depends on the energy and execution time of the mobile device (M), communication resource (C'), and computation resource (C), respectively, as mentioned in Equation (5).

$$RT = M_{ET} + C_{ET} + C'_{ET} \text{ and } E = M_E + C_E + C'_E \quad (5)$$

Reinforcement learning models in the MCC environment with the agents implement the actions. By examining the environment and the rewards or feedback, the agents make decisions for the data sequence by interacting with the environment. In the MCC scenarios, each mobile device has storage capability, battery lifetime, and processing capability constraints. Each mobile device offloads its compute-intensive tasks to the nearby cloud resources with the knowledge of the communication and computational resource capabilities. In the proposed model, the agents leverage the actions and provide rewards for the corresponding actions. It considers the reward of the individual agent in the network or cloud environment and the combined reward of all the agents while deciding on the resources in the mobile cloud, termed local and global rewards. Thus, the proposed model optimizes the resources to maximize the cumulative rewards while applying DRL in the MCC environment.

4. THE PROPOSED METHODOLOGY

The proposed method analyzes the availability and applicability of the computational and communication resources for the requested application in the mobile cloud. The approach focuses on improving the response time and minimizing DE consumption. To accomplish this, it employs the DRL model during the resource allocation in the MCC. The proposed model applies the learning algorithm and updates the strategy with the assistance of the knowledge experienced by the agent. In the proposed resource allocation optimization model, the agent stores the training data executes the learning process, and provides the decisions for the problems based on the learned knowledge from the MCC environment. Figure 2 shows the overall process flow of the ENAO-DQN methodology.

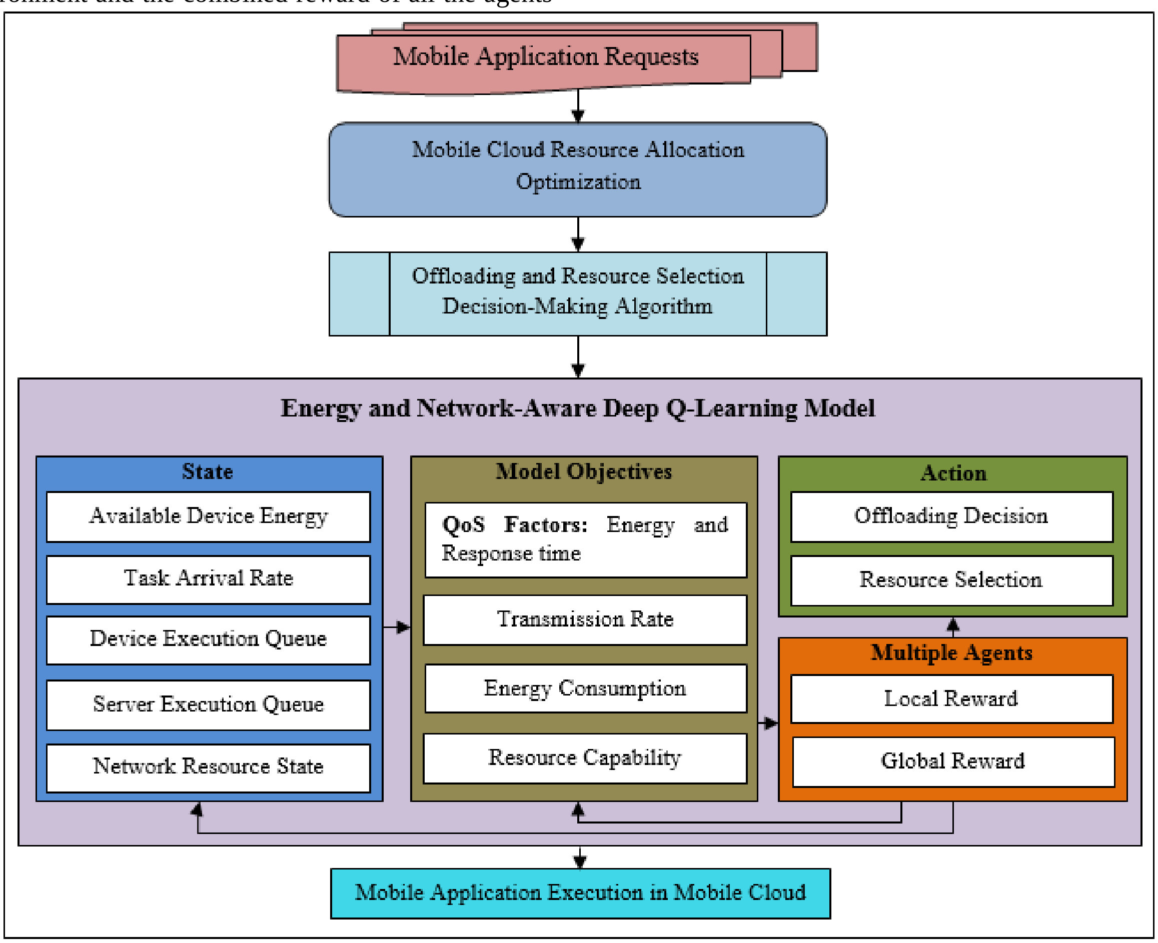

Figure 2 Energy and Network-Aware Optimization Approach in MCC

RESEARCH ARTICLE

4.1. Quality Factors for Allocation Optimization

The proposed work focuses on the quality factors such as response time and DE during the resource allocation with the modeling of the DRL. The proposed approach considers several factors as the essential parameters for the optimization process, such as the application characteristics, mobile device constraints, network resources, and cloud resources. Instead of only considering the computational resources, it examines the communication resources in the MCC environment to accomplish the QoS for the mobile users.

4.1.1. Quality Factor1: Response Time

In the proposed ENAO-DQN model, response time minimization is one of the objectives while running the mobile application requests in the cloud environment. The initial quality factor is the response time minimization that heavily relies on the network resources and the cloud computational resources due to the integrated environment of MCC. In the proposed model, the response time is based on the delay of each application involving the transmission and processing delay. The transmission delay is caused by the waiting time in queue, uploading time, internet time, and downloading time. The processing delay is the total time to process the application on mobile devices and the cloud. The proposed approach executes a subset of tasks in the device itself and the remaining tasks of the application in the cloud server based on the offloaded application complexity and the availability of resources in the mobile device. By offloading the compute-intensive requests to the remote cloud, the remote server executes the offloaded part of i^{th} mobile application using the cloud resources. Hence, the proposed resource allocation model considers the overall estimated transmission and processing time for the optimal RA. The mobile execution time (T_i^M) is based on the summation of the processing time of the tasks executed on the mobile device. The proposed approach considers multiple factors while estimating the application execution time to compute the response time (RT_i) for the estimated delay computation. The proposed method computes the response time of each application (RT_i) based on the factors of the up-link time (T_i^U), internet delay (T_i^I), waiting time in queue (T_i^Q), execution time in the cloud (T_i^C), and down-link time (T_i^D).

Knowing the estimated delay for the requested application, the ENAO-DQN model selects the appropriate network and cloud resources and analyzes the user mobility while executing offloaded applications based on the resource allocation plan. In essence, the proposed approach considers mobility as the significant parameter during the decision-making for the application execution among a set of task-resource pairs. During the network resource selection, the proposed approach estimates the transmission rate for the requested application using Equation (6). By applying Equation (6), the ENAO-DQN model selects the transmission rate, which has the minimum value according to the decision-making procedure in the reinforcement learning model.

$$\mathrm{Min}\left(R_{ik}^{T}\right)=\frac{\alpha(D_i)-\left(T_i^U+T_i^Q+T_i^I+T_i^M+T_i^C\right)-T_i^D}{U_i} \qquad (6)$$

As mentioned in Equation (6), the minimum transmission rate based network resource selection is based on the factors of the up-link time (T_i^U), waiting time in queue (T_i^Q), internet delay (T_i^I), execution time in mobile (T_i^M), execution time in the cloud (T_i^C), offloading or uploading data (U_i), down-link time (T_i^D), and threshold delay ($\alpha(D_i)$) of the i^{th} mobile application. To dynamically balance the resource allocation plan with the increased elasticity over the arrival of application requests, the ENAO-DQN model selects the network resource for the offloaded tasks based on the intermediate range. Thus, it selects the transmission rate of i^{th} application (R_{ik}^T) that has the intermediate range between minimum and maximum transmission rate of i^{th} mobile application, i.e.$\{\mathrm{Min}(R_{ik}^T) \leq (R_{ik}^T) \leq \mathrm{Max}(R_{ik}^T)\}$. The ENAO-DQN model allocates the network resource for particular application tasks (W_k^i) if the transmission rate satisfies the threshold value (α); otherwise, it reschedules the resource allocation plan based on the priority of the mobility. Hence, the proposed approach estimates the DE consumption from the execution and transmission time.

4.1.2. Quality Factor2: Energy Consumption

In addition to the network bandwidth consideration, the proposed ENAO-DQN model focuses on the essential quality factor in the mobile device, the DE. It reduces energy consumption through the mitigation of the transmission time along with the cloud execution time. In the MCC environment, the energy consumption accumulates three cases, including i) execution of applications on the mobile device, ii) application transmission and reception on the wireless network interface, and iii) application execution on the remote server. In this, the cloud resource selection or allocation depends on the effective processing speed of the resource, that is, the execution of a particular application request at minimal execution time. The estimated execution time of each request application $E_{ET}(M_i)$ on j^{th} cloud resource is based on F_i/C_j^{MIPS}. The proposed ENAO-DQN model computes that the energy of the requested application (i,j) depends on the transmission energy (E_{ik}^T), receiving energy (E_{ik}^R), and transmission rate (R_{ik}^T) of the wireless network channel. Accordingly, transmitting energy consumption (E_{ik}^T) of i^{th} mobile application on k^{th} network resource is calculated as (F_{in}/R_{ik}^T).

$$E^A(i)= E_i^M + E_{ik}^T \cdot \left(\frac{F_{in}}{R_{ik}^T}\right) + \quad E_{ij}^C + \cdot \left(\frac{F_{out}}{R_{ik}^T}\right) \qquad (7)$$

In Equation (7), F_{in} and F_{out} refer to the transmitting and receiving file or data size. Equation (7) computes the energy

RESEARCH ARTICLE

consumed by the mobile device for executing its application in the remote cloud based on the transmitting and receiving energy on the channel bandwidth along with the computation energy in the cloud.

4.2. DRL with Quality Factors

The proposed approach applies the DRL model to generate the actions as the resource allocation decision. In the proposed resource allocation optimization model, the agent obtains the reward and details about the changes in the mobile cloud environment over the dynamically increasing time steps. From the modeling of the Q-function, the proposed approach computes the reward value for each state and its corresponding action by the agent. The Q-learning agent considers four parameters in the proposed resource allocation optimization model, described in Equation (8).

$$Q_{Agent} = \{T_R, S, A, (QF_1, QF_2), K, Q\} \quad (8)$$

In equation (8), T_R denotes the requirements or features of the mobile application tasks to be allocated by the agent in the MCC environment. 'S' and 'A' refer to the state of the environment, and a set of actions belongs to the execution environment and resource selection in the mobile cloud, respectively. Quality factor1 and quality factor2 indicate the objective factors that assist the quality function based on the experience of the agent from the trained knowledge. 'Q' represents the quality function that leverages action selection for the corresponding state. The proposed approach computes and updates the Q-value based on the rewards of the current state with the maximum reward value for the state-action pairs in the mobile cloud environment.

$$\widehat{Q}(S_t, A_t) = (1-\alpha) \times Q(S_t, A_t) + \alpha \times \left(r_t + \gamma \times \left(\max_{A_{t+1}} Q(S_{t+1}, A_{t+1})\right)\right) \quad (9)$$

$$r_t = \min_k\left(R_{ik}^T\right) \text{ and } \min_j\left(E^A(i)\right), \forall\, t \quad (10)$$

In the proposed approach, the agent learns the information from each state after providing the actions for the current state over the different capabilities of the resources and different task requirements. According to Equation (3), the proposed approach assigns the reward for the agents in three aspects, including the reward of the resource capability, response time, and energy consumption. By examining the state information and the reward, the agent generates the optimal action for the resource selection. In equations (9) and (10), the reward r_t relies on the minimal response time and energy with the satisfaction of the resource capabilities derived from Equations (6) and (7). The proposed approach integrates the reward information from multiple agents located in a local state over the computation and communication resources. In the proposed model, the multiple agents are collaborated and communicate with each other locally and globally to generate the optimal actions with higher rewards.

$$\text{Optimal\_Selection}(i, j) = \operatorname*{argmax}_A Q(S_t, A_t) \&\& \operatorname*{argmax}_t r_t \quad (11)$$

The proposed approach predicts the actions for each state based on the reward patterns in the communication and computation resource selection with reference to Equation (11). In essence, the proposed DRL-based resource allocation model selects the network and computation resources that cause the maximum reward and Q-value. Moreover, the proposed approach supports the dynamic resource selection or allocation decision based on mobility, which is monitored and updated by the agent in terms of the communication time while selecting the network resources. The proposed approach learns a set of actions for the response time and energy consumption quality factors by utilizing the training knowledge. Thus, using the DRL model, the proposed approach optimally allocates the resources for the mobile applications in the MCC environment.

5. EXPERIMENTAL EVALUATION

The proposed ENAO-DQN approach is assessed with the existing Q-learning-based resource allocation optimization in the mobile cloud. Cloud offloading refers to executing all the tasks in the cloud resources, and mobile execution refers to executing all the tasks in the mobile device. Thus, the proposed ENAO-DQN is compared with three baseline models Q-learning, Cloud Offloading, and Mobile Execution.

5.1. Experimental Setup

The experimental framework employs CloudSim and its extended simulators to implement the resource allocation optimization model in the mobile cloud environment. It runs the resource allocation models on Ubuntu 16.04 LTS with an i3 processor 2.4GHz, expandable memory, and a 500 GB hard disk. The experimental model uses the Java programming language with the NetBeans IDE 8.2 version and MySQL database to execute the simulator tools. It models the simulation environment with many computational mobile devices, cloud servers, and communication RF components. The cloud server contains five hosts that can operate 5 VM with the capacity of 100 Giga Instruction Per Seconds (GIPS), and the capacity of Random Access Memory (RAM) is 32 GB with 1000 GB of secondary storage.

Furthermore, to simulate the different workloads, it considers the mobile environment with a minimum 200 number of mobile devices that increase to 2000 mobile devices over time. In the mobile environment, each mobile device executes different applications such as healthcare, video gaming, and augmented reality applications with different task sizes,

RESEARCH ARTICLE

characteristics, and task lengths. The experimental model assumes that the mobile device process 100 tasks for each time step, and the cloud server process the 2500 tasks at the same time. The experimental model assumes that the size of the tasks is between 10 and 50 MB, and downlink uplink and uplink bandwidth are set to 300 MB for the network conditions between the mobile client and the cloud server. Task length at the time of the uploading or downloading process randomly varies with the random input or output file sizes. To implement the DRL algorithm, the experimental model assigns the input size for the model depending on the number of observations, such as the capability of the mobile device, the capability of the cloud server, the capability of the network resources, and the number of remaining tasks. Moreover, the output size is based on the number of actions, such as the offloading or not and resource is selected or not.

The DRL model uses Python programming language using Tensorflow API for the experimental evaluation. In the DQN-based resource allocation optimization model, the replay memory capacity is set as 10^4. Moreover, during the training process, the discount factor is set to 0.9. The DQN model comprises five fully connected layers involving one input layer connected to three hidden layers and one output layer in which the input layer and output layers are modeled for the 'state' and 'action'. The number of training sessions or decision periods is set as 1000 and 100 testing periods with 0.1 as the learning rate for implementing the DQN algorithm. Moreover, the input or state size, output or action size, and experience replay queue size are 5, 5, and 5000, respectively.

5.1.1. Evaluation Metrics

The experimental framework employs three vital performance metrics for evaluation, such as task success ratio, energy consumption, response time, and average delay, to exemplify the performance of the mobile cloud resource allocation model.

- Task Success Ratio: The ratio between the number of completed tasks and the total number of tasks.
- Energy Consumption: The amount of consumed energy by the smartphone while executing the requested mobile application.
- Response Time: The time between the request submission in the remote server and the execution completion from the remote server.
- Average Delay: It is the average time difference between the task arrival and task departure in the waiting queue for the decision-making.

5.2. Experimental Results

The experimental model compares the resource allocation performance of the proposed ENAO-DQN with the baseline Q-learning, Cloud Offloading, and Mobile execution models while varying the scenario of the task arrival rate and average data size using four different evaluation metrics.

5.2.1. Task Arrival Rate Vs. Task Success Ratio

Figure 4 depicts the performance of the task success ratio for the proposed ENAO-DQN and the existing Q-learning, Cloud offloading, and mobile execution models while varying the task arrival rate from 5 to 25. The task arrival rate is set with the number of tasks arrived per time step. In the mobile cloud, the task success ratio linearly decreases when increasing the task arrival rate due to the increased congestion during the decision-making. The ENAO-DQN approach yields a 20.68% higher task success ratio than the existing Q-learning model, even when the task arrival rate is 25. The proposed ENAO-DQN approach significantly increases the task success ratio by optimally allocating the network and cloud resources based on the DE estimation from the transmission and execution time with the help of the deep learning model.

Moreover, it estimates the transmission rate with quality factors for the requested application to optimally select the wireless network resource to utilize the available resources. In contrast, the baseline Q-learning model fails to assess the transmission energy in the network channel while optimally utilizing the available mobile and cloud resources even when considering the rewards.

The mobile execution model fails to execute all the compute-intensive tasks in the resource-constrained mobile environment for the increased number of task arrival at a time step. Hence, the mobile execution model highly degrades its performance in the task success ratio than other comparative models. As a result, the mobile execution model yields a 45.71% minimal task success ratio than the proposed ENAO-DQN model when the task arrival rate is 25.

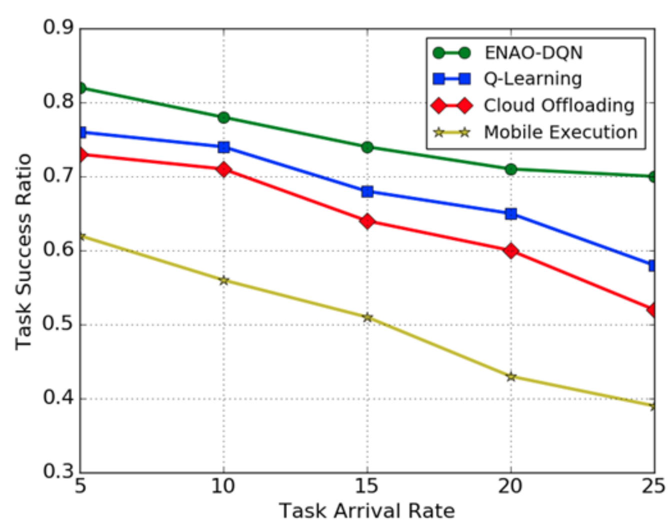

Figure 4 Task Arrival Rate vs. Task Success Ratio

RESEARCH ARTICLE

5.2.2. Task Arrival Rate Vs. Energy Consumption

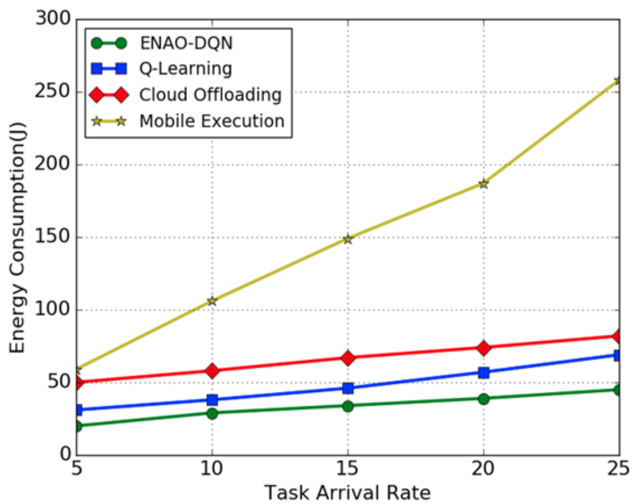

Figure 5 Task Arrival Rate vs. Energy Consumption

The DE consumption of the proposed ENAO-DQN and existing Q-learning, cloud offloading, and mobile execution models are depicted in Figure 5 when varying task arrival rates. The ENAO-DQN approach only consumes 45 joules of the energy when the task arrival rate is 25, but in the same scenario, the Q-learning consumes 69 joules of the energy. During the mobile application execution, the ENAO-DQN approach alleviates energy wastage by selecting the wireless network resource through the DQN model and considering the local and global rewards from multiple agents. The DQN-based algorithm optimally controls the execution time, communication time, resource capability, and energy during offloading decisions and resource selection. Even though the Q-learning model performs the allocation considering the rewards, the lack of modeling the multiple agents and considering the local and global rewards leads to increased energy consumption.

In addition, the Q-learning model fails to consider the quality factors of the energy and response time for the reward computation in the mobile cloud environment. Moreover, the proposed ENAO-DQN model comprises the wireless network resource selection that ensures the QoS to the mobile users. In essence, optimal task-resource pair selection in the proposed model is decided by estimating transmission time and execution time, which greatly helps achieve minimal device energy consumption.

In contrast to the mobile cloud execution, the cloud offloading and mobile execution scenarios consume 81 joules and 262 joules when the task arrival rate is 25. It is because mobile devices lack adequate resource capabilities, which leads to higher energy consumption due to the sequential execution of the tasks in the mobile processor.

5.2.3. Task Arrival Rate Vs. Response Time

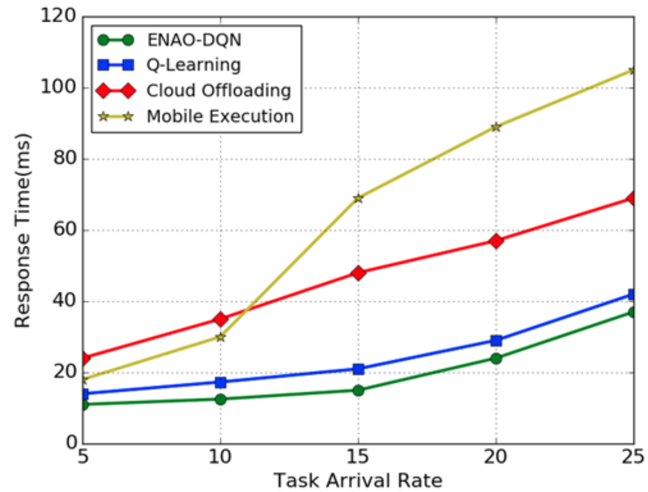

Figure 6 Task Arrival Rate vs. Response Time

Figure 6 illustrates the response time of the proposed ENAO-DQN and the existing Q-learning, cloud offloading, and mobile execution models for the different task arrival rates from 5 to 25. The response time of each request depends on the congestion rate of the task arrival in the MCC environment. When the task arrival rate is 20, the proposed approach significantly reduces response time by 33milliseconds compared to the cloud offloading model by effectively considering quality factors while selecting the network resource. In addition, the lack of modeling of the reinforcement learning for the optimal network route selection in the dynamic MCC environment leads to inaccurate decision-making for the mobile execution, cloud offloading, and Q-learning models.

Offloading all the applications to the cloud environment consumes higher communication time and leads to an extended response time of 72ms even when the cloud resources efficiently execute the task. Even though the cloud offloading ensures comparatively time-efficient than the mobile cloud execution, the optimal decision-making in the ENAO-DQN model outperforms the cloud offloading model due to the execution of lightweight tasks on the cloud consumes communication time and increases the response time to 58 ms in the cloud offloading scenario when task arrive at the rate of 20. The comparative mobile execution model consumes 68 milliseconds higher response time than the ENAO-DQN approach due to the resource-constrained mobile device lack to support for the parallel execution and the comparatively efficient execution of all the compute-intensive tasks. Even though the mobile execution model achieves the minimum response time as 19ms and 30ms when the task arrives at the rate of 5 and 10, it increases the response time upto 105ms for completing the execution of a

RESEARCH ARTICLE

higher number of tasks in the resource-scarce mobile environment.

5.2.4. Average Data Size Vs. Average Delay

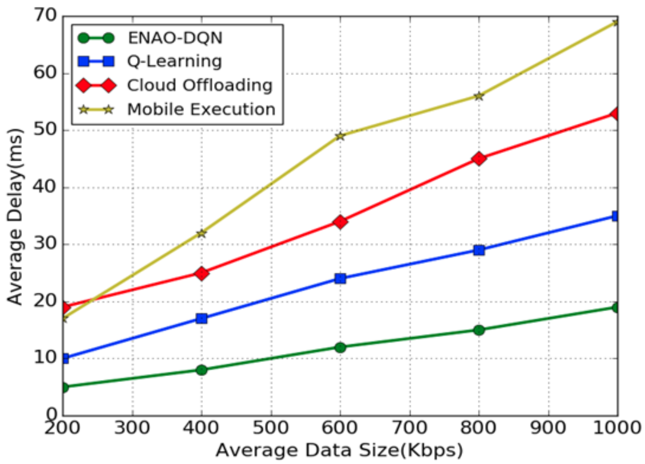

Figure 7 Average Data Size vs. Average Delay

Figure 7 compares the performance of the average delay of the proposed ENAO-DQN model with the existing Q-learning, cloud offloading, and mobile execution models. The efficiency of the decision-making algorithm greatly impacts the performance of the average delay for a set of tasks with a particular average data size. With the increase of the average data size in the mobile cloud environment, the average delay of the decision-making model increases linearly. From the results in Figure 7, it is determined that the ENAO-DQN approach reduces the average delay in executing the tasks between the departure and arrival time in the queue. The DQN-based decision-making accomplishes it for the network and cloud resource allocation over the congestion or network traffic rate in the dynamic mobile cloud environment. As a result, the proposed ENAO-DQN approach yields a minimum delay of 16ms and 34ms than the Q-learning and cloud offloading decision-making model, even when the data size is 1000Kbps. Hence, the delay becomes smaller than the Q-learning-based offloading and resource selection decision-making models. Hence, the mobile execution and cloud offloading consume 70ms and 54ms of average delay for the 1000Kbps average data size.

6. CONCLUSION

This paper presented the ENAO-DQN model, the Energy and Network-aware resource selection method based on the DRL model to reduce the consumption of DE and response time during the application execution. The proposed work considers quality factors of the mobile and application characteristics, network resource parameters, and cloud resource characteristics that assist the significant selection of the wireless network resource for the mobile user request optimally. It also achieved an effective response time using the proposed wireless network resource selection strategy concerning the quality factors. The proposed optimal resource allocation model has focused on the computation resources and communication capacity in terms of the transfer time reduction in the network channel using traffic rate, latency, and processing time-based network resource selection along with the consideration of the local and global reward. Thus, the ENAO-DQN model optimizes the resource allocation in the multi-learning agent in the MCC environment effectively to execute the offloaded tasks of the application with high QoS in terms of 34% of reduced energy consumption than the Q-learning model.

REFERENCES

[1] Fernando, Niroshinie Seng W. Loke, and Wenny Rahayu, "Mobile cloud computing: A survey", Future Generation Computer Systems, pp.84-106, Vol.29, No.1, 2013.

[2] Wu, Huaming. "Multi-objective decision-making for mobile cloud offloading: A survey", IEEE Access, Vol.6, pp.3962-3976, 2018.

[3] Zhou, Bowen, and Rajkumar Buyya, "Augmentation techniques for mobile cloud computing: A taxonomy, survey, and future directions", ACM Computing Surveys (CSUR), Vol.51, No.1, pp.1-38, 2018.

[4] Nayyer, M. Ziad, Imran Raza, and Syed Asad Hussain, "A survey of cloudlet-based mobile augmentation approaches for resource optimization", ACM Computing Surveys (CSUR), Vol.51, No.5, pp.1-28, 2018.

[5] Dinh Hoang T, Chonho Lee, Dusit Niyato, and Ping Wang, "A survey of mobile cloud computing: architecture, applications, and approaches", Wireless communications and mobile computing, Vol.13, No.18, pp.1587-1611, 2013.

[6] Noor, Talal H., Sherali Zeadally, Abdullah Alfazi, and Quan Z. Sheng, "Mobile cloud computing: Challenges and future research directions", Journal of Network and Computer Applications, Vol.115, pp.70-85, 2018.

[7] Ferrer, Ana Juan, Joan Manuel Marquès, and Josep Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing", ACM Computing Surveys (CSUR), Vol.51, No.6, pp.1-36, 2019.

[8] Parajuli, Nitesh, Abeer Alsadoon, P. W. C. Prasad, Rasha S. Ali, and Omar Hisham Alsadoon, "A recent review and a taxonomy for multimedia application in Mobile cloud computing based energy efficient transmission", Multimedia Tools and Applications, Vol.79, No.41, pp.31567-31594, 2020.

[9] Junior, Warley, Eduardo Oliveira, Albertinin Santos, and Kelvin Dias, "A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment", Future Generation Computer Systems, Vol.90, pp.503-520, 2019.

[10] Duc, Thang Le, Rafael García Leiva, Paolo Casari, and Per-Olov Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey", ACM Computing Surveys (CSUR), Vol.52, No.5, pp.1-39, 2019.

[11] Shakarami, Ali, Mostafa Ghobaei-Arani, and Ali Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective", Computer Networks, p.107496, 2020.

[12] Zhou, Guangyao, Wenhong Tian, and Rajkumar Buyya, "Deep Reinforcement Learning-based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions", arXiv preprint arXiv:2105.04086, 2021.

[13] Du, Wei, and Shifei Ding, "A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications", Artificial Intelligence Review, Vol.54, No.5, pp.3215-3238, 2021.

RESEARCH ARTICLE

[14] Zhang, Weiwen, Yonggang Wen, Kyle Guan, Dan Kilper, Haiyun Luo, and D. Wu, "Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel", IEEE Transactions On Wireless Communications, Vol.12, No.9, pp.4569-4581, 2013.
[15] Shu, Peng, Fangming Liu, Hai Jin, Min Chen, Feng Wen, Yupeng Qu, and Bo Li, "eTime:energy-efficient transmission between cloud and mobile devices", Proceedings IEEE in INFOCOM, pp.195-199, 2013.
[16] Mahinur, Fatema Tuz Zohra, and Amit Kumar Das, "Q-MAC: QoS and mobility aware optimal Resource Allocation for dynamic application offloading in mobile cloud computing", In 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp.803-808, 2017.
[17] Chen, Meng-Hsi, Ben Liang, and Min Dong, "Joint offloading and Resource Allocation for computation and communication in mobile cloud with computing access point", In IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp.1-9, 2017.
[18] Chunlin, Li, Zhou Min, and Luo Youlong, "Elastic resource provisioning in hybrid mobile cloud for computationally intensive mobile applications", The Journal of Supercomputing, Vol.73, No.9, pp.3683-3714, 2017.
[19] Chunlin, Li, Liu Yanpei, and Luo Youlong, "Energy-aware cross-layer Resource Allocation in mobile cloud", International Journal of Communication Systems, Vol.30, No.12, p.e3258, 2017.
[20] Zhang, Jing, Weiwei Xia, Feng Yan, and Lianfeng Shen, "Joint computation offloading and Resource Allocation in heterogeneous networks with mobile edge computing", IEEE Access, Vol.6, pp.19324-19337, 2018.
[21] Chen, Meng-Hsi, Ben Liang, and Min Dong, "Multi-user multi-task offloading and Resource Allocation in mobile cloud systems", IEEE Transactions on Wireless Communications, Vol.17, No.10, pp.6790-6805, 2018.
[22] Avgeris, Marios, Dimitrios Dechouniotis, Nikolaos Athanasopoulos, and Symeon Papavassiliou, "Adaptive Resource Allocation for computation offloading: A control-theoretic approach", ACM Transactions on Internet Technology (TOIT), Vol.19, No.2, pp.1-20, 2019.
[23] Alkhalaileh, Mohammad, Rodrigo N. Calheiros, Quang Vinh Nguyen, and Bahman Javadi, "Dynamic Resource Allocation in hybrid mobile cloud computing for data-intensive applications", In International Conference on Green, Pervasive, and Cloud Computing, Springer, pp.176-191, 2019.
[24] Malik, Saif UR, Hina Akram, Sukhpal Singh Gill, Haris Pervaiz, and Hassan Malik, "EFFORT: Energy efficient framework for offload communication in mobile cloud computing", Software: Practice and Experience, Vol.51, No.9, pp.1896-1909, 2021.
[25] Ali, Abid, Muhammad Munawar Iqbal, Harun Jamil, Faiza Qayyum, Sohail Jabbar, Omar Cheikhrouhou, Mohammed Baz, and Faisal Jamil, "An efficient dynamic-decision based task scheduler for task offloading optimization and energy management in mobile cloud computing", Sensors, Vol.21, No.13, p.4527, 2021.
[26] Mahesar, Abdul Rasheed, Abdullah Lakhan, Dileep Kumar Sajnani, and Irfan Ali Jamali, "Hybrid delay optimization and workload assignment in mobile edge cloud networks", Open Access Library Journal, Vol.5, No.9, pp.1-12, 2018.
[27] Li, Ji, Hui Gao, Tiejun Lv, and Yueming Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC", In 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp.1-6, 2018.
[28] Nawrocki, Piotr, and Bartlomiej Sniezynski, "Adaptive service management in mobile cloud computing by means of supervised and reinforcement learning", Journal of Network and Systems Management, Vol.26, No.1, pp.1-22, 2018.
[29] Ali, Zaiwar, Lei Jiao, Thar Baker, Ghulam Abbas, Ziaul Haq Abbas, and Sadia Khaf, "A deep learning approach for energy efficient computational offloading in mobile edge computing", IEEE Access, Vol.7, pp.149623-149633, 2019.
[30] Wang, Jiadai, Lei Zhao, Jiajia Liu, and Nei Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach", IEEE Transactions on emerging topics in computing, 2019.
[31] Eshratifar, Amir Erfan, Mohammad Saeed Abrishami, and Massoud Pedram, "JointDNN: An efficient training and inference engine for intelligent mobile cloud computing services", IEEE Transactions on Mobile Computing, 2019.
[32] Alfakih, Taha, Mohammad Mehedi Hassan, Abdu Gumaei, Claudio Savaglio, and Giancarlo Fortino, "Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA", IEEE Access, Vol.8, pp.54074-54084, 2020.
[33] Qu, Guanjin, Huaming Wu, Ruidong Li, and Pengfei Jiao, "Dmro: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing", IEEE Transactions on Network and Service Management, Vol.18, no.3, pp.3448-3459, 2021.
[34] Shakarami, Ali, Ali Shahidinejad, and Mostafa Ghobaei-Arani, "An autonomous computation offloading strategy in Mobile Edge Computing: A deep learning-based hybrid approach", Journal of Network and Computer Applications, Vol.178, p.102974, 2021.
[35] Duardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl, "MAUI: Making Smartphones Last Longer with Code Offload", ACM, In Proceedings of 8th international conference on Mobile systems, applications, and services, pp.49-62, 2010.

Authors

Dr. Puneet Sharma is currently working as Assistant Professor, Department of Computer Science & Engineering, Amity University, Uttar Pradesh, Lucknow Campus, India. He has more than 09 years of teaching and research experience. He his Ph.D. in Computer Science and Engineering from Amity University, U.P., M. Tech degree in Computer Science & Engineering from SRM University, Tamil Nadu, India, in 2013. He received his B.E. degree in Computer Science & Engineering from Annamalai University, Tamil Nadu, India, in the year 2011. In parallel he has also completed his regular Diploma certification in Networking Protocol Engineering in the year 2010 from Annamalai University itself. He has published more than 45 research publications to his credit in various Scopus, SCI, Elsevier, Scopus and IEEE international conferences. He has also published book chapters in Springer. Currently being offered a membership for London Journals Press and long-time member for IET, UK. He has hosted and organized many workshops, Faculty Development Programmes, Guest Lectures, Special Sessions in the conferences internally and externally. His research domain includes Cloud Computing, Cloud Forensics, Software testing and Operating Systems. He also has 01patent in his credit. Google Scholar Metrics: h-index: 04 with 28 citations as so far.

Dr. T. Sakthivel received his Ph.D. in Information Technology and Computer Science Engineering from MS University, Tirunelveli, Tamil Nadu, India in 2013. He has obtained B.E (CSE) from University of Madras and M. Tech (IT) from Punjabi University in 1995 and 2003 respectively. He has more than 20 years of experience in teaching, industry, and research. Currently, serving as a director at Firstsoft Technologies (P) Ltd, Chennai, and focusing on cutting-edge technologies and innovative products. His research interests mainly focus on Wireless Networks and Security, Digital Forensics, Mobile Cloud Computing, IOT, and Machine Learning. He is a professional member of the ACM.

RESEARCH ARTICLE

Prof. (Dr.) Deepak Arora is currently working as a Professor and Head, Department of Computer Science & Engineering and Information & Technology, Amity University, Lucknow Campus. He has more than 15 years of teaching, research and industry experience. He received his PhD in Computer Science from Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow in the year 2009. He has published more than 65 research publications to his credit in various Scopus, SCI, Thomson Reuters (Web of Science), ACM indexed international journals and more than 45 publications in Elsevier, Springer, ACM and IEEE international conferences. He has also published more than 30 book chapters with publishers like Springer, Taylor & Francis, ACM, and IEEE. He has also supervised 05 PhDs also. In addition to serving on the editorial, advisory and reviewer board of several international journals and conferences, he is a professional member of prestigious bodies like IEEE, ACM, IET (UK), CSI and IAENG. His research domain includes Machine Learning, Health Informatics, Distributed Computing, Object-Oriented Software Engineering, Data Mining, Big Data Analytics and Cloud Computing. Google Scholar Metrics: h-index: 10 with more than 200 citations. He has supervised 04 PhDs also.

How to cite this article:

Puneet Sharma, T. Sakthivel, Deepak Arora, “Deep Q-Learning Network-Based Energy and Network-Aware Optimization Model for Resources in Mobile Cloud Computing”, International Journal of Computer Networks and Applications (IJCNA), 9(3), PP: 361-373, 2022, DOI: 10.22247/ijcna/2022/212561.