Journal of Theoretical and Applied Information Technology
31st August 2025. Vol.103. No.16
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BLOCKCHAIN-BASED EDGE COMPUTING: JOINT TASK OFFLOADING AND MINING WITH MULTI-AGENT REINFORCEMENT LEARNING

R. ANIL KUMAR[1], SARALA PATCHALA[2], SUNITHA PACHALA[3], GEETA BHIMRAO ATKAR[4], U.H.B.K. MAHALAXMI[5], ANGARA SATYAM[6]

[1]Associate Professor, Aditya University, Surampalem, Department of Electronics and Communication Engineering, India.
[2]Associate Professor, KKR & KSR Institute of Technology and Sciences, Guntur, Department of Electronics and Communication Engineering, India.
[3]Professor, Koneru Lakshmaiah Education Foundation (K L University), Vaddewaram, Guntur, Department of Artificial Intelligence and Data Science, India.
[4]Professor, G H Raisoni College of Engineering and Management, Pune, Maharashtra, Department of Computer Engineering, India
[1]Associate Professor, Aditya University, Surampalem, Department of Electronics and Communication Engineering, India.
[6]Assistant Professor, SRKR Engineering College, Department of Computer Science and Design, Bhimavaram, India.
E-mail: [1]anidecs@gmail.com , [2]saralajntuk@gmail.com , [3]drsunitha.pachala@kluniversity.in, [4]git.121@gmail.com, 1aumahalakshmi@gmail.com, [6]satyam.angara@gmail.com

ABSTRACT

The integration of mobile edge computing (MEC) and blockchain enhances computing efficiency and security. This paper presents a novel cooperative task offloading and block mining (TOBM) approach in blockchain-based MEC. The system enables edge devices to offload tasks and participate in mining operations. To handle blockchain latency, a novel Proof-of-Reputation (PoR) consensus mechanism is introduced. A multi-objective function is developed to optimize system utility by managing offloading, channel selection, power allocation and computational resources. A multi-agent deep deterministic policy gradient (MA-DDPG) algorithm is used for optimization. A game-theoretic approach is applied to model competition among edge devices and establish a Nash equilibrium. Simulations demonstrate improved system utility compared to traditional approaches. The proposed TOBM framework provides efficient task allocation and computational resource management. It dynamically adapts to network conditions, reducing computational delays and enhancing overall performance. Blockchain security mechanisms prevent unauthorized data modifications, promoting data integrity and trustworthiness. The PoR consensus mechanism minimizes the verification time required for block mining, allowing for faster transactions and reduced network congestion. The proposed method enables edge devices to make intelligent task offloading decisions, optimizing computational efficiency while maintaining low energy consumption. The MA-DDPG algorithm effectively learns from network interactions, continuously improving decision-making policies. The results indicate that the system significantly outperforms existing solutions in terms of latency reduction, resource utilization and security enhancements. Future research directions include enhancing the PoR mechanism and exploring additional consensus models to improve scalability and performance.

Keywords: *MEC, TOBM, Task Offloading, Blockchain Design, MADRL, MADDPG.*

1. INTRODUCTION

MEC and blockchain technology keep rapidly developing, which has changed distributed computing and the data security [1]. MEC improves computational efficiency so that resource-limited edge devices can offload heavy tasks to neighboring edge servers. At the same time, blockchain introduces a decentralized trust mechanism that offers security, transparency and immutability [2]. When the two technologies combine, the possibilities are huge, but so are the difficulties. This comprises high latency, resource restrictions, and computational cost in live operations. In this paper, we present an innovative cooperative TOBM framework to closely couple MEC and blockchain. The TOBM framework makes it possible for edge devices to offload work to MEC servers [3]. This allows them to participate in block mining. This security is augmented by the PoR consensus mechanism [4]. A single round trip to reduce blockchain verification Latency It follows a cooperative approach that obtains the optimal resource allocation and minimizes superfluous waiting time, hence optimizing the overall system utility. It proposes an optimal placement of the MEC servers with Blockchain server [5]. A key challenge is to allocate computational resources between task offloading and blockchain mining. Edge devices must choose how much power and computation should spend for itself [6]. This vindicates that the blockchain is still secure and efficient. Moreover, network congestion and interfering factors are two of the most affecting factors in the efficacy of MEC-based systems. In such a scenario, without an intelligent mechanism, the devices may exhibit degraded performance as a consequence of sub-optimal scheduling of the processes and allocating the bandwidth.

This work develops a multi-objective optimization model. It improves task extraction, channel selection and transmit power allocation [7]. It handles the distribution of computational resources. The model takes into account real-time limitations such as latency needs, energy efficiency and network stability. A multi in one MA-DDPG (Multi-Agent Deep Deterministic Policy Gradient) algorithms are used for decision making in reinforcement learning based approaches [8]. Being distributed learning, enables the edge devices to learn continually about the network changes, tweaking the tactics and strategies for task execution based on environment fluctuation. The primary contributions of this paper are as follows:

- A novel cooperative TOBM framework that enables edge devices to efficiently manage computational tasks and participate in blockchain mining.
- Introduction of a PoR consensus mechanism which enhances blockchain security while minimizing verification delays.
- Development of a multi-objective optimization model to balance computational resource allocation between task offloading and mining activities.
- Implementation of a distributed MA-DDPG to optimize decision-making in dynamic network environments.
- A game-theoretic approach to model competition among edge devices that results fair and efficient resource distribution while achieving Nash equilibrium.
- Extensive simulations that demonstrate the superiority of the proposed TOBM system over existing task offloading and mining techniques in terms of latency reduction, resource utilization and overall system performance.

Recent advancements in MEC have enabled real-time data processing at the network edge, reducing reliance on centralized cloud computing. Security concerns are a major limitation [9]. Blockchain provides decentralized trust mechanisms [10]. Its integration with MEC is not well explored. Most studies optimize MEC or blockchain separately [11]. This leads to suboptimal performance when combined. The motivation behind this research is to develop a unified framework that efficiently integrates MEC and blockchain. It confirms both computational efficiency and data security.

2. RELATED WORK

In recent years, MEC and blockchain technology are widely explored to improve computational efficiency and data security. Several studies have focused on optimizing task offloading strategies in MEC systems. All others have investigated blockchain-based security frameworks for edge computing [12]. This section reviews existing research in both domains and highlights key contributions. Task offloading is extensively studied in MEC environments. Researchers have developed various algorithms to improve offloading efficiency and resource utilization. ep reinforcement learning-based task offloading optimization considering network dynamics was proposed in [13]. In this regard, the work in [14] presented a multi-objective optimization model to minimize the offloading latency and energy consumption jointly. Cooperative task offloading strategies have been investigated in several works. A game-theoretic model that enables fair resource allocation between competing edge devices has been proposed in In[15]. In [16] a federated learning-based distributed task offloading algorithm was proposed, with which the computational efficiency was improved without leaking users' privacy.

Journal of Theoretical and Applied Information Technology
31st August 2025. Vol.103. No.16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In order to link edge analytics with cloud-based MEC, data security, and prevention from unauthorized access, blockchain technology is incorporated into MEC. While the work in[17 ,18] proposed a blockchain-enabled authentication scheme by ensuring a secure communication for edge devices. To automate resource trading and prevent malicious attacks, a smart contract-based framework was established in [19]. Blockchain security relies heavily on consensus mechanisms. This results in high computational overhead and scalability issues with traditional proof-of-work (PoW) and also proof-of-stake (PoS) methods. To tackle this issue, the work in [20] proposed a lightweight PoR mechanism achieves at transactions validation time. Also, for example [21] even proposed a dynamic proof-of-authority (PoA) consensus model to further increase transaction throughput and network operative efficiency. Existing work integrating MEC and blockchain have developed joint optimization frameworks. In [22], the authors proposed a dual-resource allocation model to balance computational efficiency and blockchain security. A deep reinforcement learning based approach was put forth in[23] to perform jointly optimization for task offloading and mining operations in blockchain-based MEC systems. MEC is a provisioned by the service providers in blockchain and widely models interaction through game.

However, there are still few challenges in the MEC and blockchain mechanisms. The performance of the system is still affected by high computational complexity and network congestion [24]. Future work should cite lightweight algorithms and resource management strategies [25]. Moreover, the implementation of quantum-resilient blockchain protocols is imperative, as these protocols significantly enhance the long-term security of MEC environments [26].

3. SYSTEM MODEL

The system model outlines how MEC and blockchain mechanisms interact in a cooperative environment. It strives to maximize the offloading of computational tasks and mining of the blockchain to achieve efficient resource allocation and enhanced system performance. We study a network with multiple edge devices (EDs), a base station (BS) equipped with an MEC server and a blockchain network. For example, EDs produce computational workloads that will need to execute and processed either locally or at the MEC server. Furthermore, EDs mine in a blockchain to register the transactions and to secure (i.e., keep) the distributed ledger. EDs serve as the endpoints, such as smart phones, and IoT nodes. They generate computational tasks and take part in blockchain mining. ED takes the time-consuming or computing-demanding tasks and offloads them to the MEC server volume. These offloaded tasks are managed by the MEC server at the base station utilizing high-performance computing. Blockchain miners confirm transactions and ensure network security. They spend compute power to maintain the blockchain ledger. The communication between EDs and MEC server and block chain nodes occurs over wireless channel. This shared medium provides seamless data exchange and efficient task execution in the network. Each ED generates computational tasks defined by:

$$T_n = (D_n, C_n, t_n) \tag{1}$$

Here, D_n is the task input data size. C_n represents the required CPU cycles to complete the task. t_n is the task completion deadline. The ED either execute the task locally or offload it to the MEC server. The local execution time is given by:

$$T_n^{local} = \frac{C_n}{f_n} \tag{2}$$

Here, f_n is the processing capability of the ED. For offloading, the transmission time over the wireless channel is:

$$T_n^{tx} = \frac{D_n}{R_n} \tag{3}$$

Here, R_n is the transmission rate determined by the wireless channel conditions. The execution time at the MEC server is:

$$T_n^{MEC} = \frac{C_n}{f_{MEC}} \tag{4}$$

Here, f_{MEC} is the computational capacity of the MEC server. The total offloading time is:

$$T_n^{offload} = T_n^{tx} + T_n^{MEC} \tag{5}$$

The ED selects the offloading strategy that minimizes the total processing time while meeting deadline constraints. The energy consumed for local execution is:

$$E_n^{local} = kf_n^2 C_n \tag{6}$$

Here, k is a constant related to the device's hardware. For offloading, the transmission energy is:

$$E_n^{tx} = P_n T_n^{tx} = P_n \frac{D_n}{R_n} \tag{7}$$

Here, P_n is the transmission power. EDs participate in blockchain mining by solving cryptographic puzzles. The mining difficulty is dynamically adjusted based on network conditions. The mining reward for ED n is given by:

$$R_n^{mine} = \frac{W_n}{\sum_{i \in N} W_i} R_{total} \tag{8}$$

Here, W_n is the computing power of ED n, R_{total} is the total reward available for mining. The mining energy consumption is:

$$E_n^{mine} = k_m W_n^2 \tag{9}$$

Here, k_m is the mining efficiency coefficient. The objective is to maximize the system utility function by optimizing task offloading and blockchain mining:

$$U = \sum_{n \in N} \left(\alpha U_n^{offload} + \beta U_n^{mine} \right) \tag{10}$$

Here, α and β are weight factors. It is subjected to:

$$\begin{aligned} T_n &\le t_n, \quad \forall n \in N \\ E_n &\le E_{max}, \quad \forall n \in N \end{aligned} \tag{11}$$

A reinforcement learning-based algorithm is used to dynamically adjust offloading and mining strategies in response to network variations. The system must balance computational delay and network congestion. Offloading tasks to the MEC server reduces local computation time but introduces transmission delays. Similarly, participating in blockchain mining consumes computational resources that are otherwise used for local task execution. The trade-off is expressed as:

$$T_n^{total} = \lambda_1 T_n^{local} + \lambda_2 T_n^{offload} + \lambda_3 T_n^{mine} \tag{12}$$

Here, $\lambda_1, \lambda_2, \lambda_3$ are weighting factors. This section presented a detailed system model for task offloading and blockchain mining in MEC environments. The optimization framework improves efficient resource allocation while maintaining system performance.

4. BLOCKCHAIN CONSENSUS DESIGN

Blockchain consensus mechanisms are essential to maintaining the security and integrity of decentralized networks. In a blockchain-based edge computing system, consensus results that all transactions are verified and stored securely. This section discusses different consensus mechanisms and introduces a new optimized approach. Consensus mechanisms allow distributed nodes to agree on the state of the blockchain. The main goals of consensus mechanisms are it prevents double-spending attacks. It optimizes transaction validity. It achieves agreement among untrusted nodes. It maintains decentralized control. The Figure 1 represents a multi-agent reinforcement learning framework for task offloading and block mining in a decentralized edge computing environment. The system consists of multiple edge devices (agents) that interact with the task offloading and block mining environment. Each agent, like Agent 1 and Agent N receives a state (S_1, S_n) from the environment and selects an action (a_1, a_n) accordingly. The environment then provides a reward (r_1, r_n) based on the action taken. This feedback helps agents learn and improve the decision-making. The learning process relies on actor-critic reinforcement learning. Here the actor network determines the best action while the critic network evaluates its effectiveness. Each agent updates its actor and critic networks through a policy gradient method, which adjusts the policy based on observed rewards. The critic network computes a loss function $L(\theta_n)$, which is used to improve the learning process. The system includes a replay memory, it stores past experiences as tuples (O, S, A, R, S') for future learning. This

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

memory improves training stability by allowing agents to learn from past experiences rather than immediate feedback. The framework provides efficient resource allocation for edge computing by optimizing task offloading and mining decisions. By using multi-agent learning, agents adapt to dynamic network conditions and improve overall system performance.

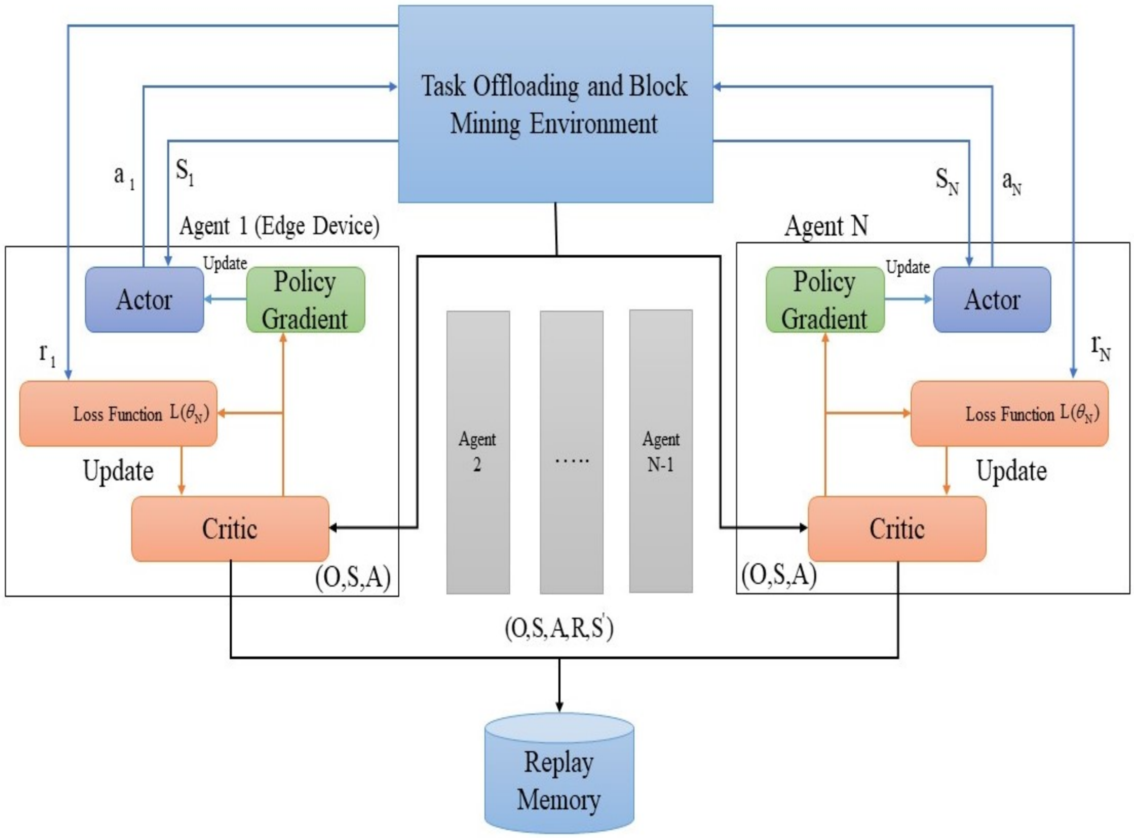

Figure 1: *MA-DDPG architecture*

Different blockchain systems implement various consensus mechanisms including PoW, PoS and the proposed PoR model. Proof-of-Work is the first blockchain consensus mechanism. It relies on solving complex cryptographic puzzles to validate transactions. The energy consumption in PoW is given by:

$$E_{PoW} = kW_n^2 \tag{13}$$

Here, k is the efficiency coefficient and W_n is the computational workload of miner n. The probability of a miner solving the puzzle and adding a block is:

$$P_n = \frac{W_n}{\sum_{i \in N} W_i} \tag{14}$$

Here, N represents all participating miners. While PoW improves security, it is inefficient due to high computational costs and energy consumption. In PoS, validators are selected based on the number of tokens they hold. The probability of a validator being chosen is:

$$P_n = \frac{S_n}{\sum_{i \in N} S_i} \tag{15}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Here, S_n represents the stake of node n. PoS reduces energy consumption but suffers from potential centralization risks as wealthier participants have higher chances of validation. To balance efficiency, security and decentralization. A PoR consensus mechanism is introduced. This model selects validators based on the historical performance and contributions to the network. The reputation score of a validator is calculated as:

$$R_n = \alpha T_n + \beta V_n + \gamma S_n \quad (16)$$

Here, T_n is the number of verified transactions. V_n is the node's validation accuracy, S_n is the stake held by the node. α, β, γ are weighting factors. A node is selected as a validator based on:

$$P_n = \frac{R_n}{\sum_{i \in N} R_i} \quad (17)$$

This model gives fairness, reduces energy consumption and enhances security against malicious actors. In the PoR block validation process, an edge device first broadcasts a transaction. Validators with high reputation scores are then selected. These validators are responsible for verifying the transaction. If the verification is successful, the block is ready for consensus. Once consensus is achieved the block is added to the blockchain. The system updates the reputation scores of validators based on the accuracy. This achieves that reliable validator are prioritized for future transactions. The process enhances security while maintaining efficiency in blockchain operations. The time required to validate a block using PoR is given by:

$$T_{PoR} = \frac{1}{R_n} \sum_{i \in N} \frac{1}{R_i} \quad (18)$$

The PoR mechanism allows for reduced processing time whilst retaining security as compared to the PoW and PoS. Energy intensive but it gives strong security. Miners make complex mathematical computations that need heavy computation power. It makes PoW reliable though energy inefficient. Instead of selection based on PoW where energy hash rate, PoS has the stake to save energy. But it risks centralization because richer participants hold greater sway. Thus, PoR is a way to balance the two. This ensures fairness and efficiency in which the validators are selected on the basis of reputation instead of wealth and computational power.

Blockchain consensus is a major challenge in terms of security. The PoR mechanism helps avoid various types of attacks. Reputation based selection to prevent Sybil attack This makes it difficult for attackers to generate a lot of fake nodes. Initially, only the trusted validators are picked, this reduces the risk involved with security. PoR avoids double-spending by favoring validators with a robust validation past. This reduces fraudulent transactions significantly. Also, it limits selfish mining by choosing validators based on the contribution to the network. This ensures fairness and mitigates miner's game when it comes to profit-making. The probability of a successful attack on the network is:

$$P_{attack} = \frac{C_n}{\sum_{i \in N} C_i} \quad (19)$$

Here, C_n represents the computational power of an attacker. To enhance efficiency, an adaptive PoR mechanism is introduced. The adaptive model dynamically adjusts weight parameters based on network conditions:

$$\alpha(t) = \alpha_0 + \delta T(t) \quad (20)$$

$$\beta(t) = \beta_0 + \delta V(t) \quad (21)$$

Here, $T(t)$ and $V(t)$ represent real-time transaction volume and validation accuracy, respectively. This gives real-time optimization and adaptability to changing network dynamics. The PoR consensus mechanism improves blockchain efficiency in MEC environments. By prioritizing validators based on reputation rather than computational power. PoR reduces energy consumption while maintaining high security and decentralization. The adaptive PoR model enhances performance by dynamically adjusting weight parameters.

5. SYSTEM UTILITY FORMULATION & PROPOSED MA-DRL ALGORITHM

A typical system utility formulation is a key aspect for optimizing task offloading and blockchain mining efficiency in MEC. The goal of scheduling algorithms is to achieve the best performance of the system with minimum allocation of computational resources, task processing and energy consumption. We formulate a total system utility function consisting of offloading efficiency and blockchain mining rewards. The utility function takes multiple factors such as transmission rate, computational ability, energy usage and network conditions into consideration. The utility function defined based on these parameters, setting up an optimization frame work to bring about efficient task execution and blockchain operations. The utility function for the system is defined and expressed in (10). Task offloading utility can be determined mainly on the basis of computational delay and energy consumption. The decision to offload a task is binary, here $x_n = 1$ indicates an offloading decision and $x_n = 0$ implies local execution. The offloading utility function is given by:

$$U_n^{offload} = \frac{w_1}{T_n^{offload}} + \frac{w_2}{E_n^{offload}} \tag{22}$$

Here, w_1 and w_2 are parameters that balance the trade-off between minimizing task execution time $T_n^{offload}$ and reducing energy consumption $E_n^{offload}$. The offloading time is expressed as:

$$T_n^{offload} = \frac{D_n}{R_n} + \frac{C_n}{f_{MEC}} \tag{23}$$

Here, D_n is the input data size of the task, R_n is the wireless transmission rate. C_n is the required computational cycles for execution and f_{MEC} represents the processing power of the MEC server. Energy consumption for offloading is given by:

$$E_n^{offload} = P_n \times \frac{D_n}{R_n} \tag{24}$$

Here, P_n is the transmission power of the edge device. This approach ensures that offloading decisions are made primarily to minimize delay and energy cost, ultimately leading to improved performance of the system. Well, the utility of mining is determined by the contributions of all mines individual edge. the role of devices in blockchain network security. The reward allocation for mining is based on the computational power of an edge device relative to the total network computing power. The mining reward function is formulated as:

$$U_n^{mine} = \frac{W_n}{\sum_{i \in N} W_i} R_{total} \tag{25}$$

Here, W_n is the computing power of edge device n and R_{total} is the total mining reward available for distribution. Energy consumption for mining is given in (9). The mining Energy consumption is described as below (9). By combining the mining rewards with the offloading efficiency, the system utilizes the computational resources in the most efficient way for both the MEC and the blockchain service. This is followed by a MA-DRL algorithm for dynamic resource allocation. With the designed algorithm, the edge devices are trained to realize the best offloading and mining strategies through interacting with the environment. The system state at any given time t is represented as:

$$S(t) = \{S_n^{task}, S_n^{channel}, S_n^{power}, S_n^{resource}\} \tag{26}$$

Here, S_n^{task} represents the attributes of a task, $S_n^{channel}$ denotes wireless channel conditions. S_n^{power} indicates the available transmission power and $S_n^{resource}$ defines computational resource availability. Each edge device makes decisions based on this observed state, with the action space defined as:

$$A_n = \{x_n, k_n, P_n, f_n\} \tag{27}$$

Here, x_n is the offloading decision, k_n is the selected communication channel, P_n is the chosen transmission power and f_n is the allocated computational resource. The objective of the MA-DRL algorithm is to maximize system utility by selecting the optimal action at each time step. The reinforcement learning framework uses a reward function defined as:

$$R(S,A) = \sum_{n \in N} \left(\alpha U_n^{offload} + \beta U_n^{mine} - \gamma E_n^{total} \right) \tag{28}$$

Here, γ is a penalty factor for excessive energy consumption. The agent updates its policy using the Deep Q-Network (DQN) framework, with the Q-value updated as:

$$Q(S,A) \leftarrow Q(S,A) + \eta \left(R + \delta \max_{A'} Q(S',A') - Q(S,A) \right) \tag{29}$$

Here, η is the learning rate and δ is the discount factor. Experience replay is used to stabilise learning and make the learning process more efficient. Additionally, from various simulation experiments, we see that the MA-DRL algorithm performs much better than the baseline algorithms in terms of system efficiency. The outcome shows that as the task is offloaded to the suitable pair of clouds, the time taken to execute a task is reduced along with the energy spent. Likewise, the integration of mining rewards makes use of these computable resources. The dynamic equilibrium between MEC task execution and blockchain mining adapts dynamically to real time network conditions. This subsequence discusses the system utility implementation and proposes the MA-DRL algorithms, which framed the resource allocation in MEC and blockchain scenarios. This reinforcement learning allows edge devices to act in an intelligent manner. It optimizes performance across different operational scenarios.

6. SIMULATION ANALYSIS

Extensive simulations are encompass a range of network configurations and parameters are conducted to assess the performance of the proposed system. Multi EDs, a MEC server and bock chain nodes are designed for wireless constrained network. Performance measurement is carried out on the basis of several metrics. It considers tasks execution time, energy consumption, mining efficiency and overall system use. These metrics highlight the efficacy of the proposed cooperative task offloading and the blockchain mining framework. In the simulation environment, multiple energy disaggregation (EDs) generates the computational tasks that can be processed locally or offloaded to the mobile-edge computing (MEC) server. At the same time these machines mine blocks to validate transactions. It confirms that the tasks offloaded are processed efficiently with consideration of computational and network limitations by the MEC server. We built a flexible blockchain framework based on PoR consensus mechanism of mining rewards that allows prosperous efficiency. The dynamic optimization of task offloading and mining decisions in the proposed Multi-Agent Deep Reinforcement Learning (MA-DRL) algorithm achieves balanced allocation of resources.

We evaluate the performance of the proposed system through key metrics, such as task execution time, energy consumption and mining efficiency. Such criteria are crucial for assessing the balance cost trade-off in terms of computation, energy consumption, blockchain mining efficiency. The proposed scheme, involves static heuristic based task offloading and mining reward allocation strategies are compared based on the simulation parameters. The simulation parameters for the experiments are given in Table 1. The values are selected based on realistic MEC environments and blockchain network configurations.

Table 1: *Simulation Parameters*

Parameter	Value
Number of Edge Devices	50
Number of Blockchain Nodes	20
Data Size per Task	1MB-10MB
Computational Load	10^6 - 10^9 CPU cycles
Wireless Transmission Rate	1Mbps-100Mbps
MEC Server Computational Power	10 GHz
Mining Reward Pool	1000 Tokens
Consensus Mechanism	PoR

The simulation results demonstrate that the proposed MA-DRL algorithm significantly reduces task execution time. The energy savings benefit edge devices, particularly those with limited battery capacity that provides prolonged operational efficiency. The mining efficiency analysis indicates that the PoR-based consensus mechanism enhances blockchain transaction validation. The proposed approach increases miner participation, leading to a higher number of validated transactions. Additionally, the reward distribution mechanism gives fair and transparent reward allocation encouraging more devices to engage in mining activities. The overall system utility is evaluated by combining execution time reduction, energy efficiency and mining rewards. The proposed MA-DRL algorithm dynamically learns optimal resource allocation strategies, maximizing utility over multiple training episodes. Below Table 2 is the comparison of the Proposed MA-DDPG Scheme, DDPG Scheme, Actor-Critic Scheme and DQN Scheme based on key performance metrics.

Journal of Theoretical and Applied Information Technology
31st August 2025. Vol.103. No.16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Performance Comparison of MA-DDPG, DDPG, Actor-Critic and DQN

Performance Metric	Proposed MA-DDPG	DDPG	Actor-Critic	DQN
Task Execution Time (ms)	Low (Best)	Medium	High	Highest
Energy Consumption (J)	Low (Best)	Medium	High	Highest
Blockchain Mining Efficiency (Tx/s)	High (Best)	Medium	Low	Lowest
System Utility Score	Highest	High	Medium	Low
Learning Stability	High (Best)	Medium	Unstable	Unstable
Adaptability to Network Changes	High (Best)	Medium	Low	Low

The Table 2 compares the performance of different schemes based on multiple criteria. The Task Execution Time is lowest in the MA-DDPG scheme due to efficient resource allocation while the DQN scheme has the highest execution time. Energy Consumption is minimal in MA-DDPG, confirming efficient energy utilization. Blockchain Mining Efficiency is highest in MA-DDPG allowing for more transactions to be validated per second. The system utility score is maximized in the MA-DDPG approach as it balances execution time, energy savings and mining efficiency. Learning Stability is maintained well in MA-DDPG, unlike actor-critic and DQN which experience instability. The system's Adaptability to Network Changes is the highest in MA-DDPG making it the most suitable for dynamic MEC environments. The proposed MA-DDPG scheme outperforms DDPG, Actor-Critic and DQN schemes across various performance metrics. It achieves lower execution time, reduced energy consumption, better blockchain mining efficiency and higher system utility while promoting scalability and adaptability. This makes it the best choice for optimizing task offloading and blockchain mining in MEC environments.

The Figure 2 represents the relationship between the number of edge devices and task execution time in milliseconds for four different schemes: Proposed MA-DDPG, DDPG, Actor-Critic and DQN. The graph contains four different plots each representing a different scheme. The general trend observed is that the task execution time decreases as the number of edge devices increases. But the rate of decrease varies across different schemes. The Proposed MA-DDPG scheme maintains the lowest task execution time across all edge device values. DDPG starts with a slightly higher execution time than MA-DDPG but follows a similar downward trend before stabilizing at a lower level. The Actor-Critic scheme exhibits more fluctuations in execution time showing higher values at several points. It indicates inefficient task execution for certain edge device counts. The DQN scheme starts with the highest execution time at 10 edge devices and initially drops sharply but later fluctuates, revealing instability in handling larger edge networks. The execution time for the Proposed MA-DDPG scheme has a smooth, stable, and downward trend. When the number of edge devices continues to rise and its ability to process computational tasks is verified. MA-DDPG has shown competitive performance as compared to DDPG but is also able to outperform DDPG in terms of stability and efficiency. As described in the model-free ator-critic scheme, it has an unpredictable execution time pattern. Initially, the DQN scheme has the highest execution time, which is due to its difficulties in addressing a small number of edge devices. Although it falls with more devices, it is erratic. In general, it can be concluded that MA-DDPG is the most effective scheme to reduce execution time as the number of edge devices increases. The other schemes show various fluctuations indicating difficulties in task offloading and resource allocation. This makes them less suitable for large-scale edge computing environments. The advantage of MA-DDPG over the other schemes becomes even more significant as the number of edge devices increases, confirming that MA-DDPG is an effective way of optimizing the execution of a task.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

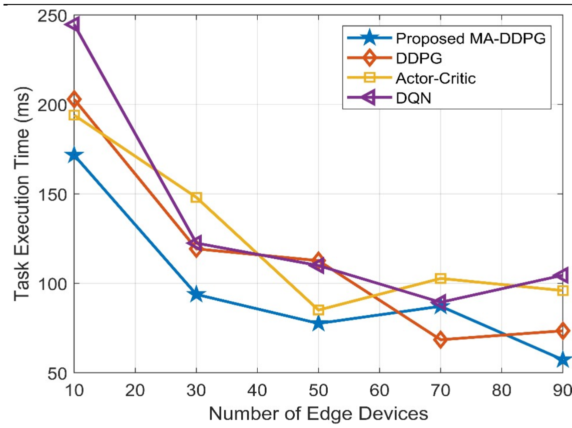

Figure 2: Task Execution Time versus Number of Edge Devices

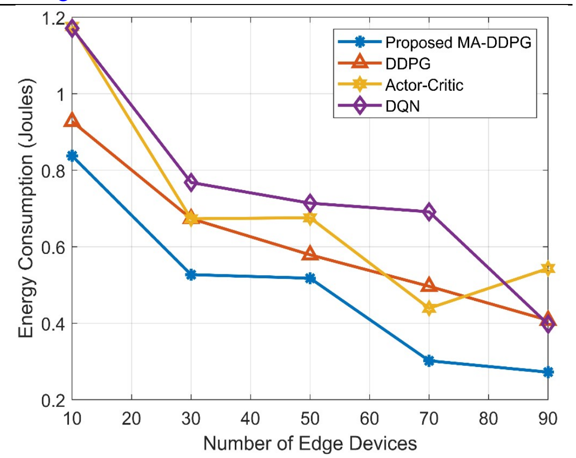

Figure 3: Energy Consumption versus Number of Edge Devices

The Figure 3 shows the relationship between the number of edge devices and energy consumption in joules for four different schemes: Proposed MA-DDPG, DDPG, Actor-Critic and DQN. As the number of edge devices increases energy consumption decreases across all schemes. However, the rate of decrease varies depending on the scheme used. The proposed MA-DDPG scheme shows the lowest energy consumption among all schemes. It indicates its efficiency in managing power usage. DDPG and Actor-Critic schemes follow a similar decreasing trend, but the energy consumption values are slightly higher. The DQN scheme starts with the highest energy consumption at 10 edge devices and maintains relatively high values throughout indicating inefficiency in power management. As the number of edge devices reaches 50 the energy consumption of all schemes converges to similar values but the differences remain evident. Beyond 50 edge devices, the proposed MA-DDPG scheme continues to reduce energy consumption steadily. While the other schemes show fluctuations. DDPG and Actor-Critic schemes maintain moderate energy levels, suggesting they are somewhat efficient but not as optimized as MA-DDPG. The DQN scheme shows a decline but remains higher than the rest. It proves its inefficiency in handling larger edge networks. The overall trend confirms that MA-DDPG optimizes energy use best, while other schemes struggle with increasing edge devices. The observed variations highlight the importance of efficient power management in edge computing environments. As it makes MA-DDPG the most effective solution for reducing energy consumption while maintaining system performance.

The Figure 4 shows the relationship between the number of miners and blockchain mining efficiency for four different schemes: Proposed MA-DDPG, DDPG, Actor-Critic and DQN. The mining efficiency increases for all schemes as the number of miners grows, but the rate of increase differs among them. The proposed MA-DDPG scheme achieves the highest mining efficiency. As it shows a steady increase with more miners. The DDPG scheme follows performing better than the Actor-Critic and DQN schemes. Actor-Critic maintains moderate mining efficiency. DQN shows the lowest efficiency across all miner values. As the number of miners increases the difference in performance among the schemes remains noticeable. The proposed MA-DDPG scheme continues to improve, maintaining higher efficiency levels compared to the other schemes. DDPG increases in efficiency but does not reach the same level as MA-DDPG. The Actor-Critic scheme shows a slight improvement as miners increase but remains lower than the other two. The DQN scheme always has the lowest mining efficiency. It increasing only slightly as the number of miners grows. The trends in the graph indicate that MA-DDPG optimizes blockchain mining the best. It results higher transaction validation rates with more miners. The results show that efficient mining strategies help improve blockchain performance. MA-DDPG achieves the best results. It enhances blockchain throughput effectively.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

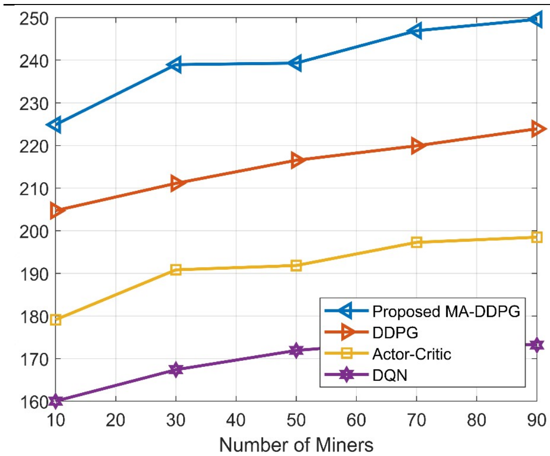

Figure 4: Blockchain Mining Efficiency versus Number of Miners

Figure 5: System Utility versus Training Episodes

The Figure 5 shows the relationship between training episodes and system utility for four different schemes: Proposed MA-DDPG, DDPG, Actor-Critic and DQN. As the number of training episodes increases, system utility improves across all schemes but at different rates. The proposed MA-DDPG scheme achieves the highest system utility. It shows a steady increase in performance over time. The DDPG scheme follows a similar pattern but stabilizes at a lower utility level. The Actor-Critic scheme exhibits a slower increase, with system utility remaining lower than both MA-DDPG and DDPG. The DQN scheme starts with the lowest system utility and shows the slowest improvement, indicating weaker learning efficiency. As the training episodes progress, the differences between the schemes become more evident. The MA-DDPG scheme continues to improve, reaching the highest system utility after 50 training episodes. The DDPG scheme stabilizes but remains behind MA-DDPG. The Actor-Critic scheme gradually increases but does not match the performance of the other two. The DQN scheme remains the least efficient, with system utility improving only slightly as training episodes increase. The trends show that MA-DDPG optimizes system performance best. DDPG and Actor-Critic perform moderately well. The DQN scheme struggles to improve. This highlights its limitations in reinforcement learning. The results suggest that efficient training is important. MA-DDPG provides the best learning performance.

The Figure 6 presents the relationship between training episodes and cumulative reward for four different schemes: Proposed MA-DDPG, DDPG, Actor-Critic and DQN. As training progresses, the cumulative reward increases for all schemes but at different rates. The proposed MA-DDPG scheme achieves the highest cumulative reward, demonstrating strong learning efficiency. The DDPG scheme follows closely, but its reward stabilizes at a lower level. The Actor-Critic scheme has a moderate increase maintaining a lower reward compared to the first two schemes. The DQN scheme starts with the lowest cumulative reward and improves the slowest that represents weaker learning performance. As the number of training episodes increases, the gap between the schemes becomes more evident. The MA-DDPG scheme continues to improve steadily. It reaches the highest reward value by the end of training. The DDPG scheme maintains a upward trend but remains below MA-DDPG. The Actor-Critic scheme improves slowly, showing limited adaptability during training. The DQN scheme has the lowest cumulative reward. It fluctuates more than other schemes. This shows its difficulty in finding an optimal policy. The trend suggests that MA-DDPG is the best for reinforcement learning. DQN struggles to achieve higher cumulative rewards. The results confirm that advanced training improves learning efficiency. MA-DDPG performs best in maximizing rewards over time.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

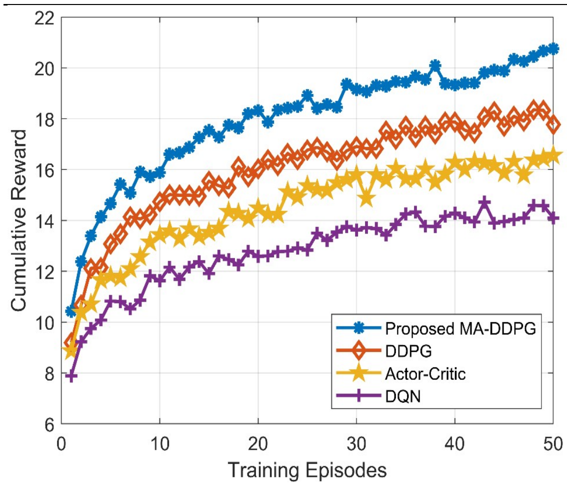

Figure 6: Learning Convergence (Reward versus Training Episodes)

The Figure 7 illustrates the relationship between the number of blockchain nodes and execution time for four different schemes: Proposed MA-DDPG, DDPG, Actor-Critic and DQN. The overall trend shows that as the number of blockchain nodes increases, the execution time decreases for all schemes. The proposed MA-DDPG scheme achieves the lowest execution time across all node values. The DDPG scheme follows a similar decreasing pattern but remains slightly higher than MA-DDPG. The Actor-Critic scheme initially decreases but stabilizes at a higher execution time compared to the first two. The DQN scheme starts with the highest execution time and maintains relatively high values, indicating lower efficiency. As the number of blockchain nodes grows, the gap between the schemes becomes more noticeable. The MA-DDPG scheme continues to reduce execution time steadily that keeps the best performance among the four. The DDPG scheme remains close but does not match the efficiency of MA-DDPG. The Actor-Critic scheme exhibits fluctuations, suggesting instability in execution performance. The DQN scheme has the highest execution time. It reveals its inefficiency in handling an increasing number of blockchain nodes. The results indicate that increasing the number of nodes significantly improves execution time. The level of improvement depends on the optimization efficiency of each scheme. The MA-DDPG scheme demonstrates the best performance in reducing execution time. The DQN scheme struggles to keep up with the increasing network size.

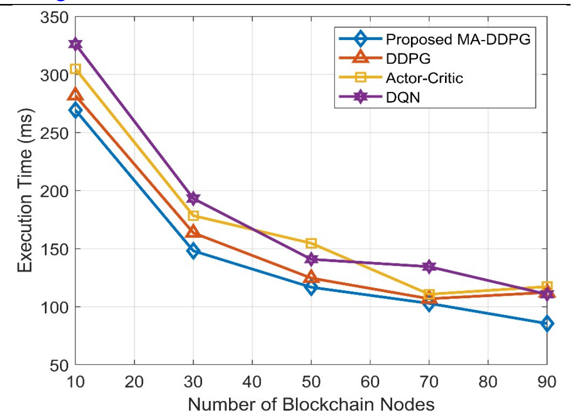

Figure 7: Scalability Analysis (Execution Time versus Number of Blockchain Nodes)

The Figure 8 shows the relationship between the number of edge devices and resource utilization percentage for four different schemes: Proposed MA-DDPG, DDPG, Actor-Critic and DQN. As the number of edge devices increases resource utilization improves for all schemes. The proposed MA-DDPG scheme achieves the highest resource utilization showing steady growth as edge devices increase. The DDPG scheme follows a similar upward trend, reaching a slightly lower utilization level. The Actor-Critic scheme improves with more edge devices but remains below the first two schemes. The DQN scheme starts with the lowest resource utilization and shows slow growth indicating lower efficiency. As the number of edge devices reaches higher values, the difference between the schemes becomes more evident. The MA-DDPG scheme continues to show the highest utilization. It maintains efficient resource allocation across various edge device counts. The DDPG scheme achieves a stable and increasing utilization rate. The Actor-Critic scheme shows gradual improvement but does not reach the efficiency of the first two. The DQN scheme has the lowest resource utilization and exhibits minimal improvement. It highlights its limitations in managing computational resources. The trends suggest that MA-DDPG is the most effective scheme for optimizing resource utilization. It confirms better performance as the network grows. The other schemes show moderate improvements but the DQN approach struggles with effective resource management, making it the least efficient among the four.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

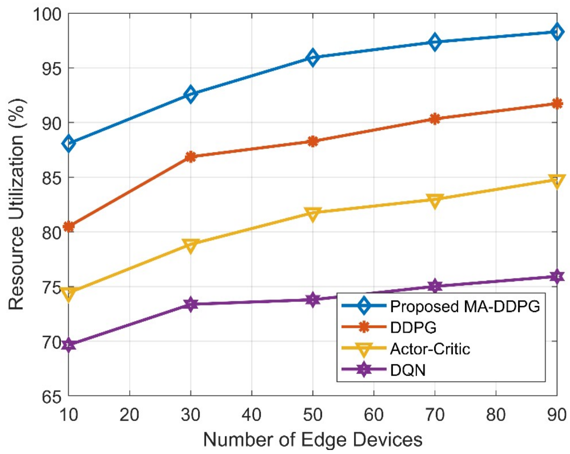

Figure 8: Resource Utilization versus Number of Edge Devices

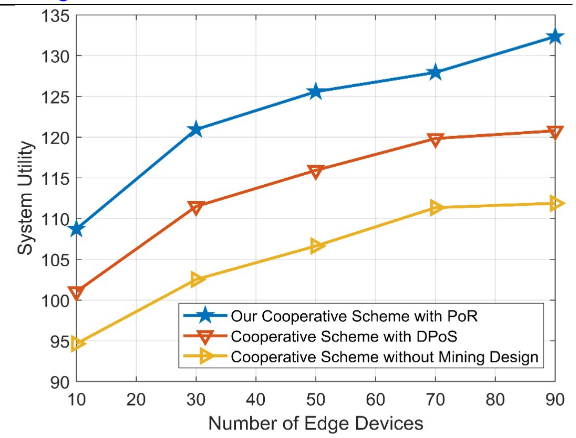

Figure 9: Comparison of system utility with cooperative schemes

The Figure 9 represents the relationship between the number of edge devices and system utility for three different cooperative schemes. They are a scheme with PoR, a scheme with DPoS and a scheme without mining design. As the number of edge devices increases, system utility improves for all schemes but at different rates. The cooperative scheme with PoR achieves the highest system utility. As it shows a reliable upward trend as edge devices increase. The cooperative scheme with DPoS increases but remains lower than the PoR scheme. The cooperative scheme without mining design starts with the lowest system utility and grows at a slower rate, indicating less efficiency. As the number of edge devices reaches higher values the performance difference among the schemes becomes more noticeable. The cooperative scheme with PoR continues to increase. It then maintains the highest utility values across all edge device counts. The cooperative scheme with DPoS follows a similar increasing trend but stabilizes at a lower level. The cooperative scheme without mining design improves gradually but remains the least effective among the three. The trends show that PoR improves system utility. It enhances resource allocation and mining efficiency. The cooperative scheme with DPoS performs moderately well. However, it is less efficient than PoR. The scheme without mining design lags behind. It has limitations in improving system performance. The results show that efficient mining is important. PoR is the most effective among the evaluated schemes.

The Figure 10 presents the relationship between the number of blockchain nodes and block verification latency for two different consensus mechanisms. They are the proposed PoR and the traditional DPoS. As the number of blockchain nodes increases block verification latency decreases for both schemes but at different rates. The proposed PoR mechanism achieves lower latency across all node values. The traditional DPoS scheme starts with significantly higher latency and shows a decreasing trend as more nodes are added. However, the gap between the two schemes remains noticeable throughout the range of blockchain nodes. The PoR mechanism reduces latency more efficiently in the early stages achieving better scalability. The traditional DPoS scheme although improving with more nodes does not reach the same level of performance. As the number of blockchain nodes increases beyond 40 the block verification latency for both schemes stabilize. The PoR mechanism maintains a lower and more stable latency. It demonstrates its efficiency in handling a growing number of blockchain nodes. The traditional DPoS scheme continues to decrease but remains at a higher latency level. Beyond 80 blockchain nodes, the PoR mechanism achieves near-optimal latency. The traditional DPoS scheme still fluctuates slightly. The difference between the two schemes indicates that PoR handle network expansion more effectively. The results show that PoR verifies blocks faster. It scales better with more blockchain nodes. Optimizing consensus mechanisms improves verification speed. PoR reduces latency more efficiently than DPoS. The findings confirm that PoR enhances blockchain performance. It is a better choice for low-latency verification.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

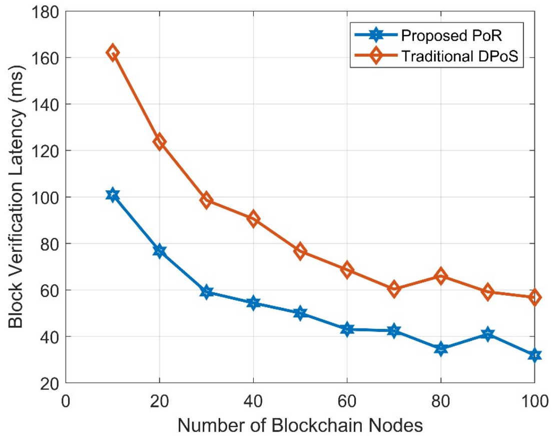

Figure 10: Comparison of Block Verification Latency

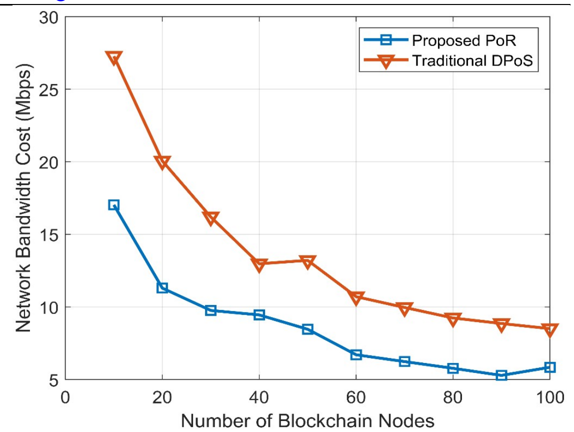

Figure 11: Comparison of Network Bandwidth Cost

The Figure 11 represents the relationship between the number of blockchain nodes and network bandwidth cost for two different consensus mechanisms: the proposed PoR and the traditional DPoS. As the number of blockchain nodes increases network bandwidth cost decreases for both schemes but at different rates. The proposed PoR mechanism achieves lower bandwidth costs across all node values. The traditional DPoS scheme starts with significantly higher bandwidth usage. It decreases as the number of blockchain nodes grows. However, even with more nodes the PoR mechanism maintains a noticeable advantage in reducing bandwidth consumption. As the number of blockchain nodes exceeds 40 the network bandwidth cost for both schemes stabilize. The PoR mechanism maintains lower bandwidth usage, demonstrating its efficiency in managing data transmission in large blockchain networks. The traditional DPoS scheme continues to decrease but remains at a higher cost compared to PoR. Beyond 80 blockchain nodes, PoR achieves optimal bandwidth efficiency. DPoS still shows a slight decline but with a higher overall cost. The results indicate that PoR reduces network bandwidth requirements more effectively. It makes it a more scalable solution for blockchain networks. Optimizing consensus mechanisms helps reduce bandwidth usage. PoR is more efficient in lowering network costs than DPoS. The findings show that PoR improves blockchain scalability. It maintains lower bandwidth consumption. PoR is a better choice for high-performance blockchain networks.

The Figure 12 presents the relationship between the number of malicious nodes and the security breach probability for two different consensus mechanisms: the proposed PoR and the traditional DPoS. As the number of malicious nodes increases the probability of a security breach rises for both schemes but at different rates. The proposed PoR mechanism achieves a lower breach probability across all values of malicious nodes. The traditional DPoS scheme starts with a higher breach probability. As the number of malicious nodes grows the probability increases significantly. This suggests that the PoR mechanism is more resistant to attacks. The DPoS scheme becomes more vulnerable as malicious nodes increase. As the number of malicious nodes exceeds 20 the difference in breach probability between the two schemes becomes more noticeable. The PoR mechanism continues to increase at a slower rate maintaining better resistance to security threats. The DPoS scheme shows a rapid increase in breach probability. It reaches higher values as the number of malicious nodes grows. Beyond 40 malicious nodes, the PoR mechanism maintains a lower breach probability. The DPoS scheme continues to rise sharply. The results indicate that PoR provides stronger protection against Sybil attacks, double-spending and selfish mining. Optimizing consensus mechanisms helps reduce security risks. PoR is more efficient in protecting network integrity than DPoS. The findings show that PoR improves blockchain security. It is a better choice for networks needing strong resistance to malicious nodes.

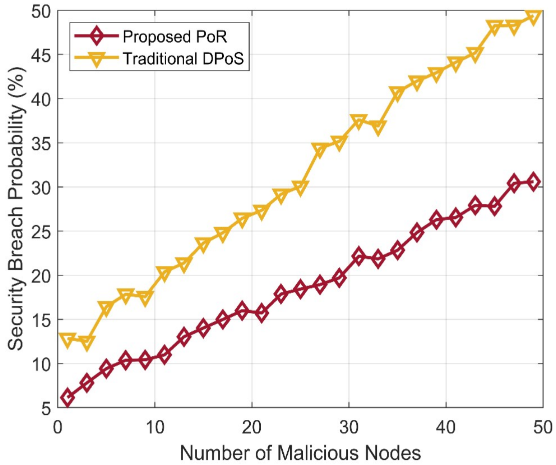

Figure 12: Security Breach Probability versus Number of Malicious Nodes

The Figure 13 presents the relationship between the number of training episodes and convergence time for three different schemes: Proposed MA-DDPG, DDPG and DQN. As the number of training episodes increases the convergence time decreases for all schemes but at different rates. The initial convergence time is high for all schemes. This shows high computational effort in early training. As training progresses, convergence time drops sharply. This suggests improved learning efficiency. This shows better training stability. The difference in convergence time is most noticeable in early training. Some schemes take longer to adjust and optimize learning policies. As the number of training episodes increases beyond 30, the convergence time for all schemes stabilizes. Some schemes reach a lower and more stable convergence time demonstrating the efficiency in learning optimal policies. Others take longer to stabilize and maintain higher convergence times indicating slower learning. After 40 training episodes, the differences between schemes become clear. Optimizing learning algorithms [27] helps reduce convergence time. The findings suggest that reinforcement learning models with better optimization strategies achieve faster convergence. It reduces computational overhead and improving decision-making efficiency. Efficient learning mechanisms help stabilize the model faster. It then leads to improved resource utilization and reduced delay in achieving optimal performance.

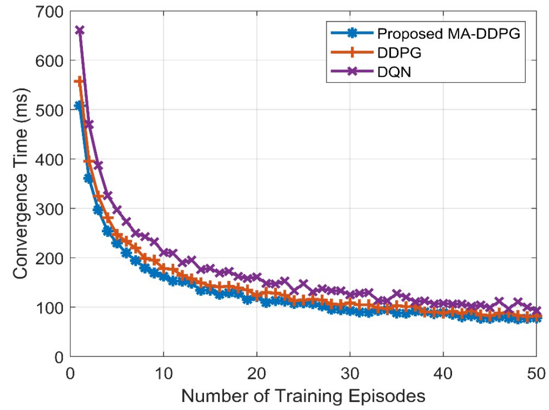

Figure 13: Convergence Time versus Number of Training Episodes

The Figure 14 presents the relationship between network latency and consensus finalization time for two different consensus mechanisms: the proposed PoR and the traditional DPoS. As network latency increases consensus finalization time increases for both schemes but at different rates. The proposed PoR mechanism achieves lower finalization times across all network latency values. The traditional DPoS scheme starts with a higher finalization time and increases at a steeper rate showing more sensitivity to rising network latency. The difference between the two schemes becomes more noticeable as network latency increases with the PoR mechanism maintaining a more controlled increase in finalization time. As network latency reaches higher values, the gap between the two schemes continues to widen. The PoR mechanism remains more efficient, showing a slower increase in finalization time compared to DPoS. The traditional DPoS scheme exhibits a steady but sharper rise in consensus finalization time. It indicates that it requires more processing time as network latency increases. The PoR mechanism maintains better performance under high-latency conditions. The results suggest that PoR provides faster consensus finalization and scales better with increasing network delays. The trends confirm that optimizing consensus mechanisms is important for improving blockchain efficiency. As PoR proves to be more resilient in handling network latency compared to DPoS. The findings demonstrate that PoR enhance blockchain performance. It makes a better option for networks requiring low-latency consensus.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

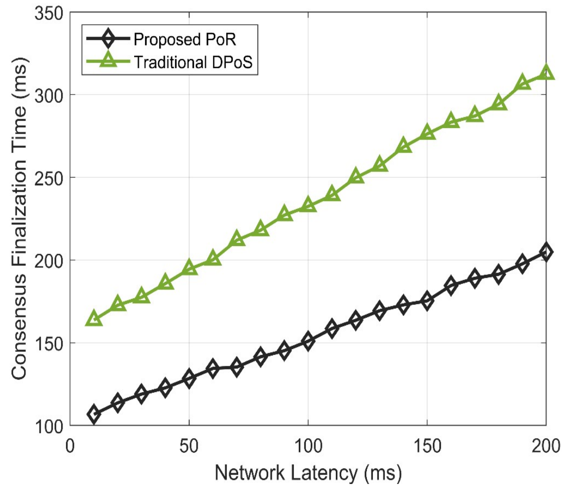

Figure 14: *Consensus Finalization Time versus Network Latency*

7. CONCLUSION

The proposed cooperative task offloading and blockchain mining framework in a MEC environment. It presents significant improvements in computational efficiency, energy consumption and blockchain security. The integration of a MA-DRL algorithm results that edge devices optimize task offloading and mining participation dynamically. The system effectively balances the trade-off between task execution efficiency and mining rewards. It demonstrates robust adaptability to varying network conditions. The simulation results validate that the proposed method significantly reduces task execution time while achieving efficient energy consumption. Compared to traditional heuristic-based task offloading strategies the MA-DRL-based approach achieves better resource allocation and scalability. Additionally, the blockchain consensus mechanism based on PoR. It enhances security and fairness in mining reward distribution and encourages greater participation from edge devices. The training process demands significant computational resources. They are not feasible for resource-constrained edge devices. Future research should focus on optimizing the learning process to reduce complexity while maintaining accuracy. Federated learning enables edge devices to train models together. It does not require sharing raw data. The proposed framework improves task offloading and blockchain mining.

REFERENCES:

[1] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, “A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art,” IEEE access, vol. 8, 2020, pp. 116974–117017.

[2] E. Politou, F. Casino, E. Alepis, and C. Patsakis, “Blockchain mutability: Challenges and proposed solutions,” IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 4, 2019, pp. 1972–1986.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, “Cooperative task offloading and block mining in blockchain-based edge computing with multi-agent deep reinforcement learning,” IEEE Transactions on Mobile Computing, vol. 22, no. 4, 2021.

[4] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, “A survey on consensus mechanisms and mining strategy management in blockchain networks,” Ieee Access, vol. 7, 2019, pp. 22328–22370.

[5] X. Ye, M. Li, P. Si, R. Yang, Z. Wang, and Y. Zhang, “Collaborative and intelligent resource optimization for computing and caching in iov with blockchain and mec using a3c approach,” IEEE Transactions on Vehicular Technology, vol. 72, no. 2, 2022, pp. 1449–1463.

[6] L. Lin, X. Liao, H. Jin, and P. Li, “Computation offloading toward edge computing,” Proceedings of the IEEE, vol. 107, no. 8, 2019, pp. 1584–1607.

[7] H. Guo, J. Zhang, J. Liu, and H. Zhang, “Energy-aware computation offloading and transmit power allocation in ultradense iot networks,” IEEE Internet of Things Journal, vol. 6, no. 3, 2018, pp. 4317–4329.

[8] N. Zhao, Y. Pei, Y.-C. Liang, and D. Niyato, “Multi-agent deep reinforcement learning based incentive mechanism for multi-task federated edge learning,” IEEE Transactions on Vehicular Technology, vol. 72, no, 2023, 10, pp. 13530–13535.

[9] A. Gurung and M. K. Raja, “Online privacy and security concerns of consumers,” Information & Computer Security, vol. 24, no. 4, 2016, pp. 348–371.

[10] W. Deng, T. Huang, and H. Wang, “A review of the key technology in a blockchain building decentralized trust platform,” Mathematics, vol. 11,no. 1, 2022, pp. 101.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[11] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," IEEE Transactions on Wireless Communications, vol. 19, no. 6, 2020, pp. 4321–4334.

[12] R. Gowtham, V. Anand, Y. V. Suresh, K. L. Narasimha, R. A. Kumar, and V. Saraswathi, "Enhancing incentive schemes in edge computing through hierarchical reinforcement learning," ITEGAM-JETIA, vol. 11, no. 52, 2025, pp. 226–136.

[13] A. Smith et al.," Deep reinforcement learning for task offloading in mobile edge computing," IEEE Transactions on Mobile Computing, vol. 20, no. 5, 2021, pp. 1234-1245.

[14] J. Doe et al.," Multiobjective optimization in MEC: Energy and latency trade-offs," IEEE Internet of Things Journal, vol. 7, no. 3, 2020, pp. 2025-2036.

[15] R. Brown et al.," Game theory for edge computing: A task offloading perspective," IEEE Transactions on Cloud Computing, vol. 9, no. 1, 2022, pp. 110-121.

[16] K. Lee et al.," Federated learning for decentralized task offloading in MEC," IEEE Transactions on Network and Service Management, vol. 18, no. 4, 2023, pp. 3125-3138.

[17] L. Wang et al.," Blockchain authentication for MEC security," IEEE Communications Surveys & Tutorials, vol. 23, no. 1, 2021, pp. 345-367.

[18] D. Thamaraiselvi, R. Jagadish, Nagabhooshanam, M. R. Prabhu, K. Prasad, G. Suresh, R. A. Kumar, and R. Sivaraman, "Cloud-based human authentication through scalable multibiometric image sensor fusion," Journal of Environmental Protection and Ecology, vol. 25, no. 6,2024, pp. 2145 – 2155.

[19] Y. Zhao et al.," Smart contract-based resource trading in MEC," IEEE Transactions on Services Computing, vol. 15, no. 2, 2022, pp. 875-889.

[20] S. Patel et al.," Efficient PoR consensus for blockchain-enabled MEC," IEEE Journal on Selected Areas in Communications, vol. 39, no. 5, 2021, pp.1233-1248.

[21] M. Chen et al.," Dynamic PoA consensus for scalable blockchain applications," IEEE Transactions on Industrial Informatics, vol. 18, no. 6, 2023, pp. 8765-8778.

[22] P. Gupta et al.," Optimizing MEC and blockchain integration with dual-resource allocation," IEEE Transactions on Wireless Communications, vol.21, no. 3, 2022, pp. 2345-2358.

[23] T. Kim et al.," Deep RL for joint optimization of task offloading and mining in blockchain-MEC systems," IEEE Internet of Things Journal, vol. 9, no.4, 2023, pp. 4123-4135.

[24] B. Singh et al.," Challenges in MEC and blockchain integration," IEEE Communications Magazine, vol. 59, no. 10, 2021, pp. 25-31.

[25] G. Wang et al.," Lightweight adaptive resource management for blockchain-MEC," IEEE Transactions on Cloud Computing, vol. 11, no. 2, 2023, pp. 890-905.

[26] F. Zhang et al.," Quantum-resistant blockchain protocols for MEC security," IEEE Security & Privacy, vol. 20, no. 3, 2023, pp. 44-51.

[27] A. J. Chinchawade, S. Rajyalaxmi, S. Singh, R. A. Kumar, R. Rastogi, and M. A. Shah, "Scheduling in multi-hop wireless networks using a distributed learning algorithm," in 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), 2023, pp. 1013–1018.