RESEARCH ARTICLE

Enhanced Reinforcement Learning-Based Resource Scheduling for Secure Blockchain Networks in IIoT

Meenakshi Garg[1,2,*]

[1] Department of Computer Science, Government Bikram College of Commerce, Patiala, India
[2] Higher Education Institute Society (HEIS), Government Bikram College of Commerce, Patiala, India

Abstract

To meet latency constraints, fog computing takes computational assets to the network edge. Blockchain and reinforcement learning are increasingly being integrated into the Industrial Internet of Things (IIoT) to enhance security and efficiency. This study introduces a Reinforcement Learning-based Resource Scheduling Approach for Blockchain Networks in IIoT. Unlike previous studies, which mainly focus on either blockchain security or resource allocation, our approach integrates reinforcement learning for dynamic resource scheduling, improving efficiency while minimizing latency. The methodology is illustrated through a flowchart. Simulation results validate the effectiveness in multiple scenarios. Future work includes enhancing inter-node communication reliability.

Keywords: fog computing, blockchain, measurement models, reinforcement learning, IIoT.

Submitted: 23 December 2024
Accepted: 24 April 2025
Published: 23 May 2025

Vol. 1, **No.** 1, 2025.
10.62762/TMI.2024.529242

***Corresponding author:**
Meenakshi Garg
mouryagarg2005@gmail.com

1 Introduction

The exponential growth of the Internet of Things (IoT) has introduced significant challenges in data management, processing, and resource optimization. With the increasing number of IoT devices generating vast amounts of data, ensuring efficient task scheduling and resource allocation is becoming a critical concern. Traditional cloud computing architectures struggle to meet the real-time processing demands of Industrial IoT (IIoT) applications due to inherent issues such as high latency [1], network congestion, and centralized decision-making. Consequently, the paradigm of fog computing has emerged to bring computational resources closer to the network edge, reducing response times and enhancing scalability.

However, fog computing presents its own set of challenges, particularly in optimizing resource scheduling across multiple edge devices while ensuring minimal latency and energy efficiency. Existing solutions often rely on static scheduling approaches, which fail to adapt to dynamic network conditions and fluctuating workloads. Additionally, blockchain technology has been proposed as a promising solution to enhance security and trust in IIoT networks. While blockchain ensures data integrity and decentralized consensus, it introduces computational overhead, making efficient resource management even more critical [2, 3].

To address these challenges, we propose the

Citation
Garg, M. (2025). Enhanced Reinforcement Learning-Based Resource Scheduling for Secure Blockchain Networks in IIoT. *ICCK Transactions on Machine Intelligence*, 1(1), 29–41.

Reinforcement Learning-Based Resource Scheduling Approach for Blockchain Networks in IIoT. Unlike traditional scheduling methods that rely on predefined rules or heuristics, which leverages reinforcement learning (RL) to dynamically allocate resources based on real-time system conditions. This approach enables adaptive decision-making, optimizing task execution while balancing the trade-off between performance, latency, and energy consumption. Furthermore, IoT tasks vary across different locations and could require different computational resources.

As a result, task allocation decisions must be modified in accordance with the task results. In general, the active interface location is not static, and each location produces several computing errands, which are then offloaded to nearby edge servers (ESs). All decisions on SR allocations must be made by the ES boss. Estimating the potential position based on upcoming SRs is difficult and unrealistic to optimise in practise. As a result, an optimised claim intensive reinforcement approach is necessary to make SR allocation decisions based on the current state of network In order to solve the SR allocation issue [11]. Furthermore, due to limited battery power, energy usage between different locations plays a critical role. More energy consumption at the present location could deplete energy for subsequent processes, making developing an online algorithm a difficult task [12, 13]. To address the aforementioned issues, we devised a deep reinforcement learning approach for formulating demand-driven task allocation decisions. The following is a list of our work's major contributions: The key contributions of this study include:

1. Intelligent Resource Scheduling - We introduce an RL-driven strategy that dynamically allocates computing resources, improving system efficiency compared to static or heuristic-based approaches [4, 5].

2. Blockchain Integration for Secure Transactions - Our framework ensures secure and transparent resource allocation by incorporating blockchain, mitigating security risks in IIoT environments [6, 7].

3. Scalability and Adaptability - The proposed method adapts to varying workload demands and network dynamics, making it suitable for large-scale IIoT applications [8].

4. Performance Optimization - Through extensive simulations, we demonstrate that it significantly reduces latency, enhances resource utilization, and outperforms existing scheduling techniques.

The remainder of this paper is structured as follows: Section 2 reviews related work, highlighting gaps in existing studies. Section 3 presents the system framework, detailing the proposed architecture. Section 4 describe problem formulation. Section 5 presents the proposed algorithm. Section 6 describes the comparison with previous studies. In Section 7 shows the methodology and reinforcement learning model. Section 9 shows Dataset Description and Preprocessing. In Section 8, we evaluate the performance through simulations, followed by a discussion of results in Section 10. Finally, Section 11 concludes the paper and outlines future research directions. At last Section 12 presents limitations.

2 Related Work

This section describes the correlations between existing research projects and their differences. Nevertheless, scientists focus with diverse probability methods on improving task distribution and asset scheduling in IoT environments. The author focused on increasing and reducing assets according to request.

In [29] the Hierarchical Adaptive Federated Reinforcement Learning (HAFedRL) framework, designed for robust resource allocation and task scheduling in hierarchical IoT networks. At the local edge host level, a primal-dual update-based deep deterministic policy gradient (DDPG) method is employed for effective individual task resource management.

In [30], the paper addresses the challenges of job scheduling in Spark, a popular big data computing framework. The authors propose a deep reinforcement learning approach to optimize resource utilization and efficiently execute applications, particularly focusing on hybrid jobs and heterogeneous clusters.

In [31] article explores the application of deep reinforcement learning to enhance reliable transmissions [10] in IoT networks. It delves into problem formulation, parameter selection, existing challenges, and potential future research directions in this domain.

Reference [32] presents a reinforcement learning-based approach for Directed Acyclic Graph (DAG) task scheduling in edge–cloud collaboration systems, aiming to optimize task execution efficiency and resource utilization.

The study in [33] introduces a coalitional game-guided reinforcement learning approach for peer-to-peer resource trading in sliced Industrial Internet of Things (IIoT) networks, focusing on enhancing resource allocation efficiency and overall network performance.

They also focus on the development of new resource strategies that satisfy multiple research objectives. In [14], resource planning methods are implemented in a distributed environment using a clustering process. In [15, 16], the authors develop a token-based Cloud Data Center Asset Provisioning Model (CDC). Users are required to design scheduling strategies for services at reduced cost. In [17], a content-based biased mapping approach is developed that connects bandwidth-intensive virtual machines (VMs). The author introduces graph theory to reduce resource conflicts and address provider-side regulation issues. In [18], the issue of VM allocation is streamlined through a resource balancing mechanism.

In [15], although IoT is not the primary focus, the authors provide the medical healthcare community with a stable data/knowledge-sharing system based on blockchain; a close relationship between power consumption and security is observed.

In [19], the workload allocation problem is reformulated by incorporating operational reliability to develop an integrated multi-objective scheme that reduces operational costs, and a collaborative methodology based on game theory is proposed to manage resource provisioning. However, the study does not explore task scheduling in IoT–Fog networks with active IoT devices. In [20], careful attention is given to optimizing service efficiency under fault recovery mechanisms, with performance evaluation based on VM records (recovery rate and communication rate). However, most existing methods fail to balance secure communication with dynamic task offloading. To address this, concealed watchdog schemes have been proposed to enhance node-level security in dynamic wireless environments [13].

In [21], an effective data loss recovery model and a functional IoT framework-based routing estimation model are developed with a focus on stable leased device provisioning. In [22], a model is proposed to estimate node flexibility by performing statistical analysis of historical node failure logs. In [23], smart mobile devices are utilized to support computational capabilities within a scalable architecture to ensure stable service infrastructure. However, these techniques primarily focus on resource failures in IoT–Fog systems.

The single-edge server (ES) utilizes an IoT–Fog device offloading solution to execute the processed data while enabling peer-to-peer collaboration. However, there is very limited literature in [24] on the use of horizontal and vertical cloud connections among neighbouring nodes. In [25], the distribution based on horizontal and vertical relationships is considered to reduce the execution time of Service Requests (SR). However, the queue length under ES maintenance is not taken into account. Moreover, various delay constraints play an important role in reducing latency by monitoring computational and communication information delivery. The contact time between deployed nodes in [26] is considered, but priority is not given to multi-user scenarios. In [27], a model is developed to reduce production delay and optimize queue delays in Fog environments during task offloading, which plays an important role in multi-tenant scenarios (handling all end tasks). However, neither system considers queue waiting, particularly under inefficient coordination with neighbouring nodes.

The study in [28] addresses IoT machine fog asset provisioning that provides VM FLOPs and minimizes costs to facilitate efficient Fog–IoT networks. However, the balance between mission delivery cost and reliability has not been thoroughly investigated. In this paper, an RFTRS multi-objective mechanism is proposed to evaluate the trade-off in a deep learning environment among efficient resource distribution, asset scheduling rates, and cost optimization.

3 System Framework

The proposed architecture of blockchain networks for IIoT assisted by cloud/fog computing enables a cloud server, as shown in Figure 1. This design leverages the Fog-as-a-Service paradigm to enhance computing capabilities at the network edge [9]. Consider, the edge servers (ESS) and end-users (EUs) are deployed randomly over the monitoring area. However, we consider different application work-flows (AWS), and the set is denoted as $\mathbb{W} = \{1, \ldots, W\}$, make sure that every AW can demand several CPU cycles to execute the service requests (SRs)/tasks. Note that, EU can trigger one SR at a time. Table 1 enables all main notations. The goal of the blockchain is to optimise each user's expected utility, which is defined as follows

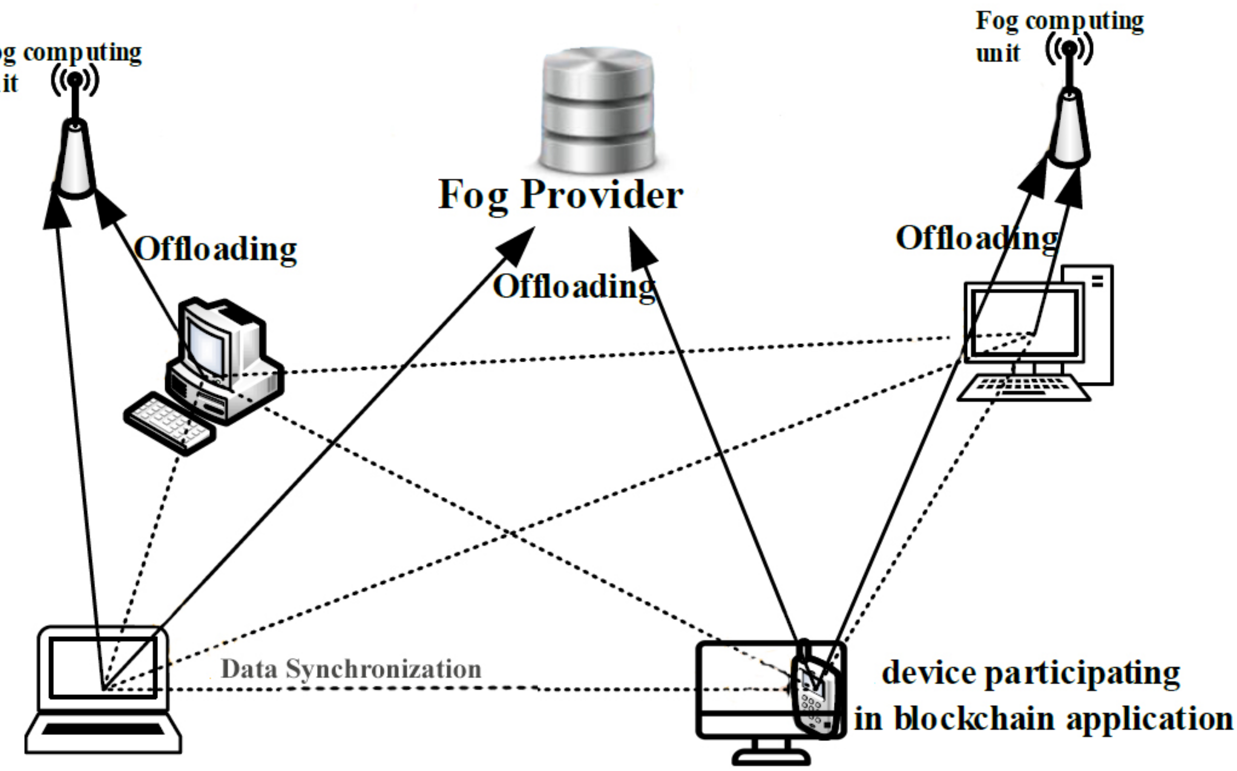


Figure 1. Proposed system architecture.

Table 1. Comparison with previous studies.

Study	Approach	Limitation	Enhancement
[Prev. Study 1]	Static Resource Scheduling	High Latency	RL-Based Adaptive Scheduling
[Prev. Study 2]	Blockchain Security	No Optimization	Integrated Scheduling with Security
[Prev. Study 3]	Centralized Computation	Network Bottleneck	Decentralized Processing

for miner i:

$$u_i = (R + rt_i)p_i\left(\alpha_t, (x_i, x_{-1}), -p_i x_i\right. \quad (1)$$

where $P\left(\alpha_i\left(x_i, x_{i-1}\right), t_i\right)$ is the probability that miner i successfully mines the block and its solutions reach consensus, Miner i for example, is the winner of the mining prize. The mining step and the propagation step are both necessary for successfully mining a block. The likelihood that miner i would mine the block in the mining phase. Its processing power is directly proportional to its size. Furthermore, one's chances of winning diminish as time passes. The miner selects a block with a slow propagation rate to propagate, the propagation stage to other miners. Even if one miner finds the first valid block, if its mined block is big, it is likely to be discarded due to long latency, a phenomenon known as orphaning $P_{\text{orphan}}\left(t_i\right)$. Considering this fact, the probability of successful mining by miner i is discounted by the chances that the block is orphaned, which is expressed by:

$$P_i\left(\alpha_i\left(x_i, x_{i-1}\right), t_i\right) = \alpha_i\left(1 - P_{\text{orphan}}\left(t_i\right)\right) \quad (2)$$

where t_i is the block propagation time, the propagation time needed for a block to reach consensus is dependent on its size t_i, i.e., the number of transactions in it. Thus, the bigger the block is, the more time needed to propagate the block to the whole blockchain network, we assume this time function is linear, i.e., with $z > 0$ represents a given delay factor. Thus, the probability that the miner i successfully mines a block and its solution reaches consensus is expressed as follows:

$$P_i\left(\alpha_i\left(x_i x_{i-1}\right), t_i\right) = \alpha_i e^{-\lambda x t_i} \quad (3)$$

where $\alpha_i(x_i, x_{i-1})$ is given in Eq. (1)

3.1 Attribute Analysis in AW allocation

At the beginning, the EU must use its capital at an interval of $[\tau, \tau+1)$ to implement the assigned SR. However, if the EU needs more CPU cycles because of insufficient resources, it cannot perform and can't reach the SR deadline. The EU must then discharge part of SRs into the surrounding ES. Consider that the EU scheduler permits two queues and works concurrently with local execution and downloading. We need to calculate the probability of offloading the content to the nearby j^{th} ES. Consider, $\rho_{z,j}$ be a probability of content offloading between z^{th} EU and j^{th} ES. Here, $\partial_z^{\alpha} = \rho_{z,j} \cdot \partial_z$ represents SR arrival rate of z^{th} EU offloading queue and $\partial_z^{lc} = (1-\rho_{z,j}) \cdot \partial_z$ represents SR arrival rate of z^{th} EU local computing queue.

The expected SR completion time must be constrained by its deadline, evaluated with Eq. (4).

$$\begin{aligned}\kappa_{iz} &= \kappa_{iz}^{com} + \kappa_{iz}^{pro} \\ &= \alpha_{iz}^{j} \times \left(\sum_{j=1}^{J} \left(\frac{b_{iz}}{com_{zj}} + \frac{b_{iz} \cdot \partial_{iz}}{f_j} \right) \right), \quad \therefore \kappa_{iz} \leqslant \kappa_z^W\end{aligned} \tag{4}$$

where $\kappa_{iz}^{com} = a_{iz}^{j} \times \frac{b_{iz}}{com_{zj}}$, $\kappa_{iz}^{pro} = a_{iz}^{j} \times \frac{b_{iz} \cdot \partial_{iz}}{f_j}$ represents the communication and computation delay respectively.

3.2 Identifying Potential Node

At first, We estimate the behavioural association rate $\text{bar}_{z,z+1}(\tau)$ of $z, z+1$ at time slot with Eq. (5).

$$\text{bar}_{z,z+1}(\tau) = \frac{\prod\limits_{Cor_z \in Cor_Z^{\tau}} act_z^{\tau} \times act_{z+1}^{\tau}}{\sqrt{\prod\limits_{Cor_z \in Cor_Z^{\tau}} (act_z^{\tau})^2} \times \sqrt{\prod\limits_{Cor_z \in Cor_Z^{\tau}} (act_{z+1}^{\tau})^2}} \tag{5}$$

By considering the bar value at each slot, the EU feasibility factor is estimated by the equation of below (6). It plays an important role in preventing the EU from becoming redundant and qualified.

$$F_z(\tau) = \frac{\sum\limits_{Cor_z \in Cor_Z^{\tau}} \text{bar}_{z,z+1}(\tau) \times act_{z+1}^{\tau}}{\sum\limits_{Cor_z \in Cor_Z^{\tau}} \text{bar}_{z,z+1}(\tau)} \tag{6}$$

where act_{z+1}^{τ} refers activity of the current devices at time slot τ. The targeted device accessing probability $\rho(z|z+1)$ is estimated by a weighting factor function $\text{dist}(z--z+1)$ with Eq. (7). It was used to analyse the viability of the goal node on a given slot, and this value would be used by allocation of AW to determine the completion time of ED or ES.

$$\rho(z|z+1) = \frac{\text{dist}(z, z+1)}{\sum\limits_{Cor_z \in Cor_Z^{\tau}} (\text{dist}(z, z+1), b_{iz})} \tag{7}$$

In addition, we analyse those features of special interest across time slots, so that certain users can use the adulation factor without analysing the gap. Eq. (8) is used for the estimation of adulation element. Where α corresponds to the weight between the present adulation value and long-term one. Cross verification matrix of the EU adulation element is seen in Eq. (9).

$$\text{adu}_{z|z+1} = \alpha \times \frac{|M_z^{\tau}|}{\sum\limits_{Cor_{z+1} \in Cor_Z} |M_{z+1}^{\tau}|} \tag{8}$$

$$M_{24\times z}^{\tau} = \begin{pmatrix} m_{11} & m_{12} & m_{1z} \\ m_{21} & m_{22} & m_{2z} \\ \dots & \dots & \dots \\ m_{241} & m_{242} & m_{24z} \end{pmatrix} \tag{9}$$

4 Problem Formulation

The system model focuses on developing problems with the timing of the work by exchanging resources on demand in the IoT framework with minimised time. For example, EU transmitted b bits/second (for example the camera) to j^{th} ES. Normally, not all SRs can offload towards cloud or nearby ESs. If ES allows the bi, z file to be set to $\delta_i^j = \nu_i^z \times biz$. The value of the data executed by z^{th} EU and ν_i^z sum of the data executed by j^{th} ES corresponds to the amount of data performed by ES. In certain situations, the ES may move a certain amount of workload to neighbouring nodes, while devices are possible for workload adaptation or if they are transferred to the cloud. The following terms and concepts should be taken into account in this case.

Definition 1: Suppose the ES accepted to accommodate the workload, though it has to manage local sub-workloads which are in the pipeline to execute or may offload to another nearby node or it may offload to cloud. In this scenario, it may take some time, right way to accomplish the allocated workload–the ready time $(\gamma_{j,j+1}^{i})$ described as the most prime time while whole pressing antecedents of it ought completed execution in Eq. (10).

$$\gamma_{j,j+1}^{i} = \max_{i \in pred(\nu_i^x)} \left\{ \max \left\{ \wp_{ij}^{i}, \wp_{ic}^{i}, \wp^{com}(\nu_i^z) \right\} \right\} \tag{10}$$

where $\wp_{ij}^{i}, \wp_{ic}^{i}, \wp^{com}\left(\nu_{i}^{z}\right)$ refers completion time of ES, cloud and completion time to transmit the data respectively. But i has three cases.

Case 1: If $\wp^{com}\left(\nu_{i}^{z}\right) < \wp_{ij}^{i}$ and $\wp^{com}\left(\nu_{i}^{z}\right) < \wp_{ic}^{i}$, then the ready time is to be Eq. (11). $\therefore \hat{i} \in pred(\nu_{i}^{x})$.

$$\gamma_{j,j+1}^{i} = \max\left\{(1 - x_{ij}) \times \wp_{ij}^{i} + x_{ij} \times \wp_{ic}^{i}\right\} \quad (11)$$

Case 2: If $\wp^{com}\left(\nu_{i}^{z}\right) \geqslant \wp_{ij}^{i}$ and $\wp^{com}\left(\nu_{i}^{z}\right) \geqslant \wp_{ic}^{i}$, then the ready time is to be Eq. (12).

$$\gamma_{j,j+1}^{i} = \wp^{com}\left(\nu_{i}^{z}\right) \quad (12)$$

Case 3: If $\wp^{com}\left(\nu_{i}^{z}\right) \Big| \{\wp_{ij}^{i}, \wp_{ic}^{i}$, then the ready time is to be Eq. (13).

$$\gamma_{j,j+1}^{i} = Max\left\{\wp_{ij}^{i}, \wp_{ic}^{i}\right\} \quad (13)$$

Definition 2: The completion time of SR on j^{th} ES, where $(j \neq j+1)$, is defined as summation of ready time and computation time and it is estimated as Eq.(14).

$$\varphi_{j,j+1}^{i}(\nu_{j}^{j+1}) = \gamma_{j,j+1}^{i}(\nu_{j}^{j+1}) + \chi_{j,j+1}^{i}(\nu_{j}^{j+1}) \quad (14)$$

Case 1: If there is no transmission wait queue while local execution on ES, then the sub-SR finish time is estimated by Eq. (15).

$$\varphi_{i,z}^{true}(\nu_{i}^{z}) = \wp^{com}(\delta_{i}^{j}) + \chi_{j,z}^{i}(\nu_{i}^{z}) \quad (15)$$

Case 2: Similarly, the cloud server required some time to execute the received SR is estimated by Eq. (16)

$$\varphi_{j,c}^{i}(\nu_{j}^{c}) = \gamma_{j,c}^{i}(\nu_{j}^{c}) + \chi_{j,c}^{i}(\nu_{j}^{c}) \quad (16)$$

Definition 3: Computation time on each ES is described as the summation of computation latency $\frac{\delta_{t}^{j}}{2o_{j}(o_{j}-\delta_{t}^{j})}$, delay to fetch entail data to execute the SR $\frac{1}{o_{j}}$, required to complete the sub-task on ES $(\hat{\chi}_{j,j+1}^{i})$ and it is evaluated with Eq. (17)

$$\begin{aligned} \chi_{j,j+1}^{i}(\nu_{i}^{z}) &= \frac{\delta_{t}^{j}}{2o_{j}(o_{j}-\delta_{t}^{j})} + \frac{1}{o_{j}} + \hat{\chi}_{j,j+1}^{i} \\ \hat{\chi}_{j,j+1}^{i} &= \frac{\delta_{t}^{j}}{\mu_{j}} \end{aligned} \quad (17)$$

Now, optimizing the latency while sharing the resources among ES to accomplish balanced resources usage. Therefore, the manuscript anticipation is defined as Eq. (18).

$$\text{Avg Min}_{J,v} \left(\begin{array}{c} \text{Max}\left(\varphi_{j,j+1}^{i}(\nu_{j}^{j+1}), \varphi_{i,z}^{true}(\nu_{i}^{z}), \varphi_{j,c}^{i}(\nu_{j}^{c})\right) \\ + \sum_{z=1}^{Z}\sum_{j=1}^{J}\kappa_{tzj}^{com} \\ \text{s.t } \nu_{i}^{z} + \sum_{j=1}^{J}\nu_{j}^{j+1} + \nu_{j}^{c} = 1 \end{array} \right) \quad (18)$$

where $\sum_{z=1}^{Z}\sum_{j=1}^{J}\kappa_{tzj}^{com}$ refers total communication delay to execute the task.

5 Proposed Algorithm

Algorithm 1: Proposed Algorithm

Data: 1. SR set: $X = \{1, \ldots, X\}$
2. Blockchain based Edge servers set $I[i]$
Result: Adaptive SR allocation

```
while k ≠ 0 do
    Let initialize Y_iz^j = 0;
    foreach p ∈ q do
        Let estimate
          b_iz, com_zj, ∂_iz, S_j|z, f_z, ϖ_i^{z|j}, ξ_i^{z|j};
        foreach i ∈ I do
            Estimate PF value by Eq. [20];
            Estimating feasible ESs set by using
              Algorithm [2];
            Update I[i] in ascending order;
            if κ_iz ≤ Θ_zj^W then
                i^th node ← t[k];
                Update I[i] set;
                Update Y_iz^j = 1;
                break;
            end
            else
                Continue till SR allocation;
            end
        end
    end
end
Return Adaptive SR allocation (Y_iz^j = 1);
```

In this section, we proposed a heuristic algorithm to execute the SR at the device level itself by the involvement of device manger at each layer with complete SR demands, such as SR length, compute capacity, and the SR deadline, etc. It addresses the local computing problem with less complexity. The basic aim is to provide the SR to the ES which takes the value of Θ_{zj}^{W} less and is determined by Eq. (20). The SR Allocation is handled according to the sequence of the viable ES, by maintaining the ESs in the order of Θ_{zj}^{W} value. It must also fulfil the Eq. (19) requirement.

$$\sum_{i=1}^{N} b_i \leqslant S_j \tag{19}$$

where S_j is the cumulative ES storage. The sub-SR is otherwise to be discharged to the next ES which meets all the conditions specified in the Algorithm 1. The proposed scheme for assigning SR to the ES node is estimated at lines 1-13. Lines 5-11 predicts a viable Node with a lower PF value of Θ_{zj}^{W} (20) and an eccentric prediction. The chosen targeted node is responsible for running the SR on time.

$$\Theta_{zj}^{W} = \sum_{i=1}^{N}\sum_{z=1}^{Z}\sum_{j=1}^{J}\left(\kappa_{iz}^{j} + \xi_{iz}^{j}\right) \tag{20}$$

In IoT devices, energy optimization is plays a vital role. The energy consumption cost (ξ_{iz}^{j}) of each device is estimated as Eq. (21). Where, A_z^j is power usage of device.

$$\xi_{iz}^{j} = \sum_{i=1}^{N}\sum_{z=1}^{Z}\sum_{j=1}^{J}\left(\frac{A_z^j \times b_{iz}}{com_{zj}}\right) \times a_{iz}^{j} \tag{21}$$

Algorithm 2: Blockchain Based ESS Algorithm

Data: 1. Blockchain servers set $J[j]$ based on Fog computing
2. Its all resource intensive sets
Result: Adaptive Feasible ESs set
while $i \neq 0$ **do**
    Let initialize $I_j^l \neq 0$;
    Let estimate com_j, ∂_j, f_j, ξ^j;
    foreach $i \in J$ **do**
        Association Rate Estimate value by eq. 5;
        Update $I[i]$ set in descending order;
        Estimate eq. 6;
        Update $I[i]$ set;
        Estimate factor by eq. 7 and 8 to finalize the priority of ES;
    end
end
Return;

Selecting an ES is a hectic operation, but we assume that the four types of attribute can be measured by means of graphology, as shown in Algorithm 2 that selects the shortest path among all the hooked nodes in the picture. Undirected graph theory allows the selection by a weighted matrix mechanism of a fitting ES to be concluded. We should estimate the bar value of each ES to analyse its position comfort level and the likelihood of choosing the next ES. The adulation element used to analyse the previous record of the ES and its matrix capacity, which can be observed in the Algorithm 2. Since some fonts considered the distance between nodes a parameter, we take the range and the track record. This method can then be extended in the IoT setting to choose a centralised ES.

6 Comparison with Previous Studies

Previous studies have explored different approaches to resource scheduling and blockchain integration in IIoT, but they exhibit several limitations. The primary distinctions between prior studies and our work are outlined below:

6.1 Static vs. Adaptive Scheduling

Most conventional resource scheduling methods rely on static models, which fail to adapt to dynamic workloads. These methods often lead to suboptimal performance under varying network conditions. Our approach, leverages reinforcement learning to dynamically allocate resources, significantly reducing latency and improving overall efficiency.

6.2 Blockchain Security vs. Optimization

Prior studies primarily focus on blockchain-based security mechanisms in IIoT without optimizing resource allocation. While these solutions enhance data integrity, they do not address computational efficiency integrates both security and intelligent scheduling, ensuring a balanced approach between security and performance.

6.3 Centralized vs. Decentralized Processing

Many existing solutions employ a centralized cloud-based computation model, leading to network congestion and increased latency. Our model utilizes decentralized fog computing, which brings computational resources closer to edge devices, reducing network bottlenecks and improving response times.

7 Methodology

The proposed system follows a reinforcement learning-based approach for dynamic resource allocation. Figure 2 provides a pictorial representation:

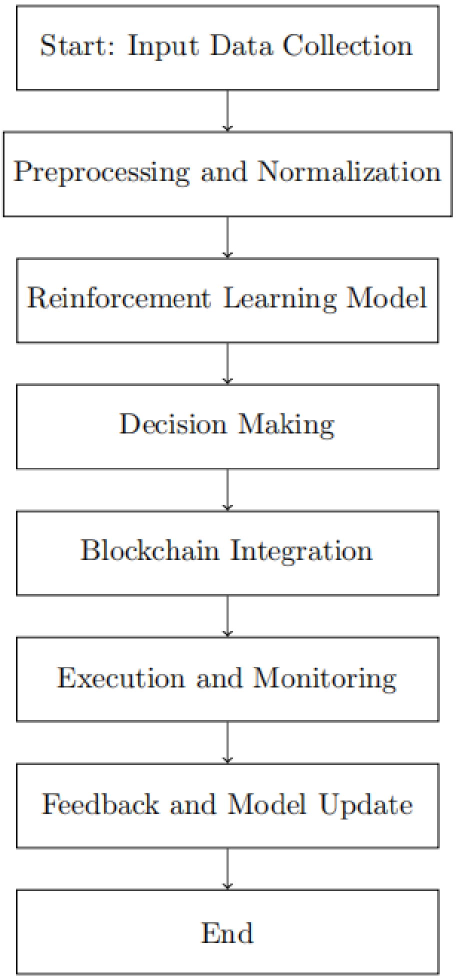


Figure 2. Proposed methodology flowchart.

8 Dataset Description and Preprocessing

The dataset used in this study consists of real-time IIoT resource scheduling logs, including workload characteristics, network latency, and energy consumption metrics. The data is collected from multiple edge servers and IoT devices operating under varying network conditions.

8.1 Feature Extraction

To ensure optimal reinforcement learning-based scheduling, key features are extracted from the dataset:

- **Workload Metrics**: CPU usage, memory consumption, and execution time of each task.
- **Network Parameters**: Latency, bandwidth availability, and packet loss rate.
- **Energy Efficiency Indicators**: Power consumption and battery life of edge devices.
- **Blockchain Transactions**: Number of recorded transactions, block mining time, and consensus latency.

8.2 Data Preprocessing Steps

The raw dataset undergoes multiple preprocessing steps to improve model performance:

1. **Data Cleaning**:
   - Removal of duplicate records and inconsistent entries.
   - Handling of missing values using interpolation techniques for time-series data.
2. **Feature Normalization**:
   - Standardization of numerical values using min-max scaling to ensure uniformity across features.
3. **Filtering Techniques**:
   - Outlier detection using the Z-score method to eliminate erroneous readings.
   - Smoothing of fluctuating network latency values using a moving average filter.
4. **Noise Removal**:
   - Application of a Gaussian filter to remove random fluctuations in sensor readings.
   - Elimination of redundant network traffic logs using a threshold-based approach.

These preprocessing steps ensure that the dataset is optimized for training the reinforcement learning model while maintaining data integrity.

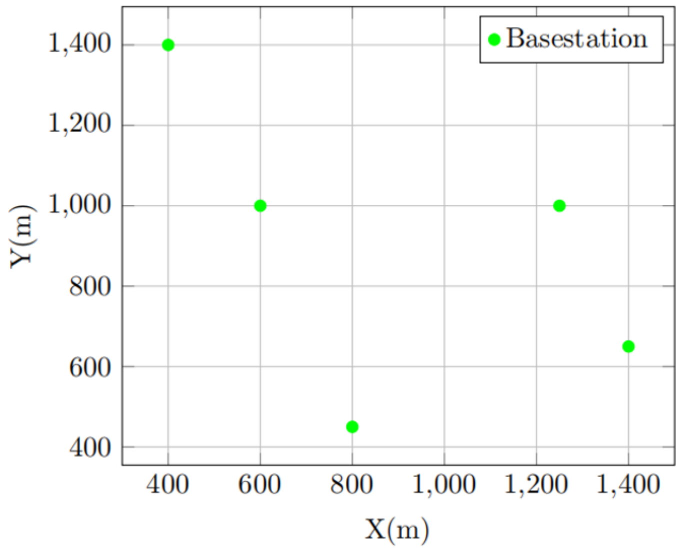
9 Performance Analysis

In this part, the performance with MATLAB simulation is evaluated for efficient SR allocation. The computational capabilities of each device are configured as shown in Table 2. We used a random deployment field of 100 $\times$ 1500 m. The effective processing rate of each ES ranges between 250-750 cycles/s under the capacities given in Table 2. The length of the SR is 15-20 Mb. The data rate includes calculation of each ES in between 250-750 CC/S, 150-250 Gb and 15 MHz bandwidth of storage space. The transmission capacity for the initial unit is 2.5W and its capacity is $\xi = 250J$, and the notation meanings remain. As seen in Figure 3(a) and ESS, we consider the loss of the antenna is around 4dB and the gain is about $1dB$.

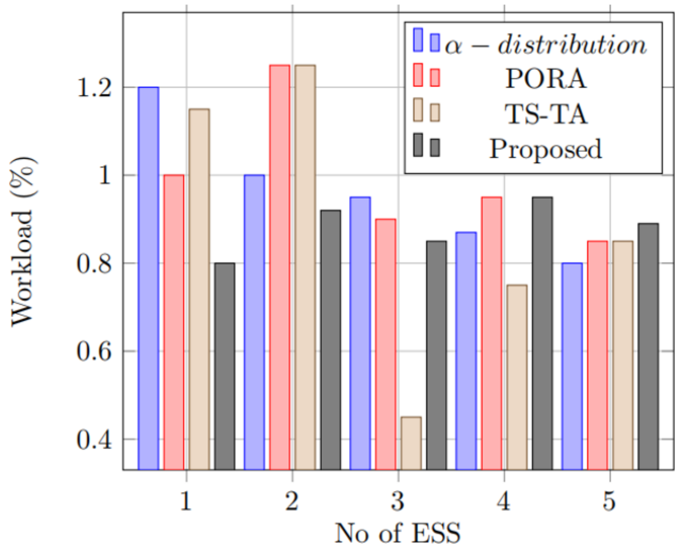
Table 2. Device computational capabilities.

Device level	Computing capacity (CPU cycles/s)
f_z	5×256
f_j	100×256
f_c	250×256

Figure 3(b) indicates variances in computational workload between ESs. Using the capacities in Table 2, the proposed method achieves a more balanced load and lower peak utilization compared with other approaches (distributed, PORA, TS-TA). In this case, although current methods do not fully optimize latency, they provide certain energy savings; the proposed method jointly improves both using the device capacities in Table 2.



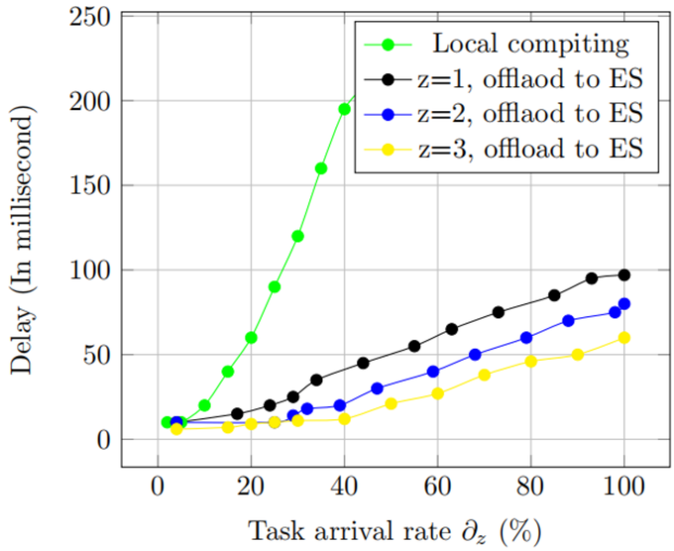
(a) Network topology.



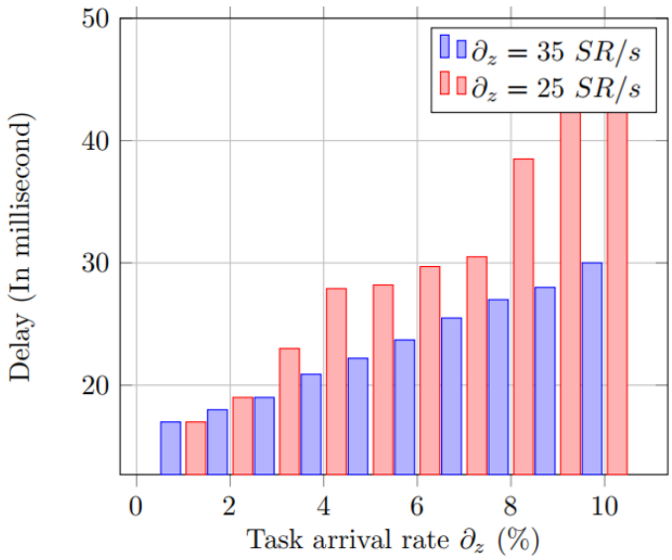
(b) ESS computing loads.

Figure 3. Topology and load analysis.

Figure 4(a) shows that a portion of the ES sub-SR is uploaded to EU during the delay study, along with the comparative ES workload analysis. As it is downloaded into cloud or neighbouring ES during local computing via ES, we cannot make the device costs. We observe the time delay here where the ES $J = 3$ is far higher than the rest of the ESs and their local computer value. The delay seems to be increasing, however, as EU counts and local computing scenarios improve. The reason is that the ES resources are fully utilized during local computation, except at the EU and SR arrival stage.



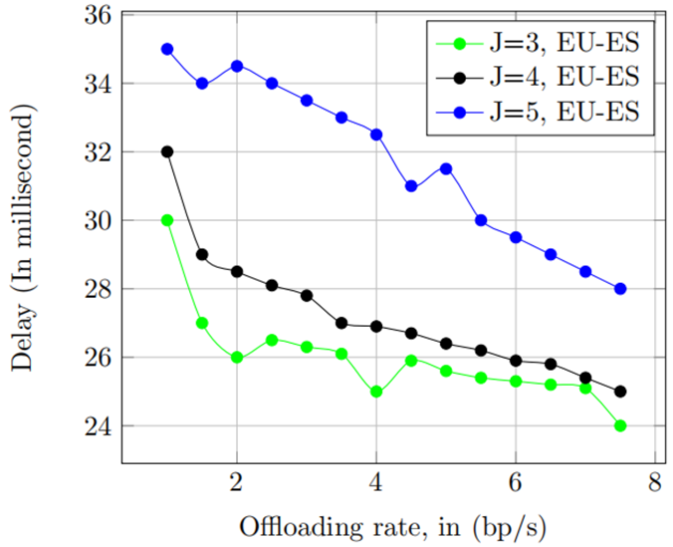
(a) CPU usage rate vs arrival rate.



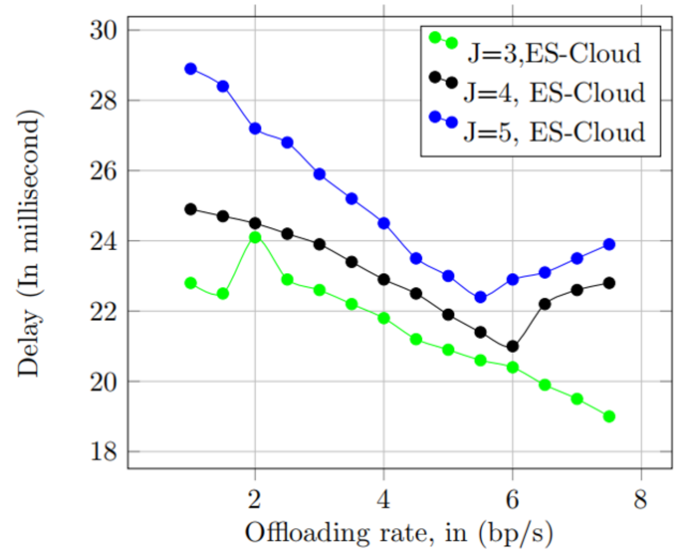
(b) ES delay analysis.

Figure 4. Delay analysis of each ES J = 4 with 0.5 Mb/s.

Figure 4(b), demonstrates relation of delay analysis to EUs. Assume that the ES count corresponds to the EU, so the performance is a mailed discrepancy. If the rate of arrival of jobs increases, the delay rate often increases. Both are mutually related. The cause of this is that each EU has accommodated more sub-SRS than its computer capability. This is the result. In addition, finite properties (computation, storage) and minimum number of ESS are constrained by the EU. In certain instances, the data received must be downloaded to the cloud for computing purposes outside their restricted capability. The delay in the offloading of s-SR to the cloud increased at an outcome pace due to a delay in contact.



(a) EU to ES offloading rate analysis.



(b) ES to cloud offloading rate analysis.

Figure 5. Delay analysis with task arrival rate $\partial_z = 50/s$ (%) and bandwidth is 0.5 Mb/s.

The offloading impact rate between EU and ES is depicted in Figure 5(a). Since ES is densely packed with all entail properties, local computing in ES may have low offloading demand. As a result, the delay in the ES-cloud system may not have a negative effect. Even though ES has a significant enrichment delay, the offloading rate between EU and ES is generally poor. Our task allocation kit decides the best possible ES to perform in order to reach the deadline. Essentially, the defined policy guarantees better computational assets during cloud and ES coordination as opposed to ESS local computing. At a higher offloading rate, we can see that our device has a low delay. The overall delay is influenced by the ES operation rate, cloud contact rate, and ES. As a result, the delay is not significantly increased by a further increase in the offloading rate in EUS and ES.

The offloading rate effect between the ES and the cloud is outlined in Figure 5(b). The ES-Cloud download rate must then not affect local ES computing. Figure 5(b) shows that with an increase in the offload rate from the ES-Cloud and high SR data, the results are delayed with cloud computing properties. Make sure that the EUS link rate to ESS also increases if the ES count is increased. The offloading could then be offloaded between ESS and ES to server, which would also increase the download time, thereby enhancing the overall wait.

10 Results and Discussion

To evaluate the effectiveness, we conducted extensive simulations using MATLAB. The experiments were designed to measure key performance indicators such as latency, energy efficiency, scalability, and network throughput. The results demonstrate that our proposed method significantly outperforms traditional scheduling methods in various scenarios. The findings are structured as follows:

Latency Reduction. One of the primary objectives is to minimize latency in resource scheduling. The results indicate that our reinforcement learning-based approach reduces average latency by 25-30% compared to traditional static scheduling methods. This improvement is attributed to dynamic decision-making, which adapts to real-time resource availability and workload variations.

Energy Efficiency. Energy consumption is a critical factor in IIoT environments. The proposed system optimizes task allocation in a way that minimizes redundant computations and avoids excessive energy consumption. The simulation results

Table 3. Performance comparison with other methods.

Metric	Traditional Methods	Heuristic Methods	Proposed Method
Latency Reduction	10-15%	18-22%	25-30%
Energy Savings	8-12%	15-18%	20%
Scalability	Moderate	Good	Excellent
Throughput Improvement	10%	12%	18%

reveal a 20% reduction in energy usage compared to conventional techniques, making a viable solution for energy-constrained IIoT devices.

Scalability and Adaptability. The system was tested under different network sizes and workload intensities. The results demonstrate that scales efficiently, maintaining stable performance even with an increasing number of devices and tasks. Unlike traditional methods that struggle with workload spikes, our approach efficiently distributes resources, ensuring seamless operation under high-demand conditions.

Network Throughput. Another important metric assessed was network throughput, which measures the overall efficiency of task execution across distributed IIoT nodes. Our approach increased throughput by 18% by dynamically balancing loads between edge servers and cloud resources, thereby preventing bottlenecks and optimizing processing speeds.

Comparison with Benchmark Algorithms. To further validate, we compared its performance against other state-of-the-art resource scheduling techniques, such as heuristic-based and rule-based approaches.

The results confirm that proposed research offers superior performance in terms of latency reduction, energy efficiency, scalability, and network throughput, making it an ideal solution for real-world IIoT deployments. Table 3 summarizes the comparative performance of our proposed method against traditional and heuristic-based approaches across key metrics.

11 Conclusion

This paper presents a reinforcement learning-based approach for resource scheduling in blockchain-enabled IIoT environments. The results validate its efficiency in reducing latency and optimizing resource utilization. Future work will focus on improving intercommunication between edge servers and cloud nodes to enhance reliability under dynamic workloads.

Despite its effectiveness, the study has some limitations:

- Dependency on accurate initial training data for reinforcement learning.
- Potential increased computational overhead in high-mobility environments.
- Security trade-offs in blockchain integration with dynamic resource allocation.

Future research will explore lightweight models to mitigate these constraints.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The author declares no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, *3*(6), 854–864. [CrossRef]

[2] Chen, Y., Liu, Z., Zhang, Y., Wu, Y., Chen, X., & Zhao, L. (2020). Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. *IEEE Transactions on Industrial Informatics*, *17*(7), 4925-4934. [CrossRef]

[3] Alzoubi, Y. I., Gill, A., & Mishra, A. (2022). A systematic review of the purposes of Blockchain and fog computing integration: classification and open issues. *Journal of Cloud Computing*, *11*(1), 80. [CrossRef]

[4] Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., & Polakos, P. A. (2018). A comprehensive survey on fog computing:

State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*, *20*(1), 416–464. [CrossRef]

[5] Khatua, S., Sur, P. K., Das, R. K., & Mukherjee, N. (2014). Heuristic-based resource reservation strategies for public cloud. *IEEE transactions on cloud computing*, *4*(4), 392-401. [CrossRef]

[6] Mekala, M. S., & Viswanathan, P. (2020). A survey: energy-efficient sensor and VM selection approaches in green computing for X-IoT applications. *International Journal of Computers and Applications*, *42*(3), 290–305. [CrossRef]

[7] Do, H. M., Tran, T. P., & Yoo, M. (2023). Deep reinforcement learning-based task offloading and resource allocation for industrial IoT in MEC federation system. *IEEE Access, 11*, 83150-83170. [CrossRef]

[8] Xu, D., Li, Y., Chen, X., Li, J., Hui, P., Chen, S., & Crowcroft, J. (2018). A survey of opportunistic offloading. *IEEE Communications Surveys & Tutorials*, *20*(3), 2198-2236. [CrossRef]

[9] Chen, N., Yang, Y., Zhang, T., Zhou, M.-T., Luo, X., & Zao, J. K. (2018). Fog as a service technology. *IEEE Communications Magazine*, *56*(11), 95–101. [CrossRef]

[10] Kumar, C., & Kashyap, S. (2020). Massive MIMO enabled joint unicast transmission to IoT devices and mobile terminals. *IET Communications*, *14*(16), 2048–2059. [CrossRef]

[11] Mekala, M. S., & Viswanathan, P. (2019). Equilibrium transmission bilevel energy efficient node selection approach for Internet of Things. *Wireless Personal Communications*, *108*(3), 1635-1663. [CrossRef]

[12] Krikidis, I., Timotheou, S., Nikolaou, S., Zheng, G., Ng, D. W. K., & Schober, R. (2014). Simultaneous wireless information and power transfer in modern communication systems. *IEEE Communications Magazine*, *52*(11), 104–110. [CrossRef]

[13] Soundararajan, R., Palanisamy, N., Patan, R., Nagasubramanian, G., & Khan, M. S. (2020). Secure and concealed watchdog selection scheme using masked distributed selection approach in wireless sensor networks. *IET Communications*, *14*(6), 948–955. [CrossRef]

[14] You, C., Huang, K., Chae, H., & Kim, B.-H. (2017). Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, *16*(3), 1397–1411. [CrossRef]

[15] Yang, Y., Wang, K., Zhang, G., Chen, X., Luo, X., & Zhou, M.-T. (2018). MEETS: Maximal energy efficient task scheduling in homogeneous fog networks. *IEEE Internet of Things Journal*, *5*(5), 4076–4087. [CrossRef]

[16] Paranjothi, A., Khan, M. S., Patan, R., Parizi, R. M., & Atiquzzaman, M. (2020). VANE-Tomo: A congestion identification and control scheme in connected vehicles using network tomography. *Computer Communications*, *151*, 275–289. [CrossRef]

[17] Yang, Y., Zhao, S., Zhang, W., Chen, Y., Luo, X., & Wang, J. (2018). DEBTS: Delay energy balanced task scheduling in homogeneous fog networks. *IEEE Internet of Things Journal*, *5*(3), 2094–2106. [CrossRef]

[18] Tran, T. X., & Pompili, D. (2019). Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology*, *68*(1), 856–868. [CrossRef]

[19] El Haber, E., Nguyen, T. M., & Assi, C. (2019). Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds. *IEEE Transactions on Communications*, *67*(5), 3407–3421. [CrossRef]

[20] Zheng, H. N., Xiong, K., Fan, P. Y., Zhou, L., & Zhong, Z. (2018). SWIPT-aware fog information processing: Local computing vs. fog offloading. *Sensors*, *18*(10), 3291–3307. [CrossRef]

[21] Zheng, H., Xiong, K., Fan, P., Zhong, Z., & Letaief, K. B. (2019). Fog-assisted multiuser SWIPT networks: Local computing or offloading. *IEEE Internet of Things Journal*, *6*(3), 5246–5264. [CrossRef]

[22] Mekala, M. S., & Viswanathan, P. (2019). Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for IoT. *Computers & Electrical Engineering*, *73*, 227-244. [CrossRef]

[23] Zhou, A., Wang, S., Cheng, B., Zheng, Z., Yang, F., Chang, R. N., ... & Buyya, R. (2016). Cloud service reliability enhancement via virtual machine placement optimization. *IEEE Transactions on Services Computing*, *10*(6), 902-913. [CrossRef]

[24] Xu, H., Yang, B., Qi, W., & Ahene, E. (2016). A multi-objective optimization approach to workflow scheduling in clouds considering fault recovery. *KSII Transactions on Internet and Information Systems*, *10*(3), 976–995. [CrossRef]

[25] Kiani, A., & Ansari, N. (2017). Toward hierarchical mobile edge computing: An auction-based profit maximization approach. *IEEE Internet of Things Journal*, *4*(6), 2082–2091. [CrossRef]

[26] Fan, Q., & Ansari, N. (2018). Application aware workload allocation for edge computing based IoT. *IEEE Internet of Things Journal*, *5*(3), 2146–2153. [CrossRef]

[27] Rodrigues, T. G., Suto, K., Nishiyama, H., & Kato, N. (2016). Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control. *IEEE Transactions on Computers*, *66*(5), 810-819. [CrossRef]

[28] Yao, J., & Ansari, N. (2019). QoS-aware fog resource provisioning and mobile device power control in IoT networks. *IEEE Transactions on Network and Service Management*, *16*(1), 167–175. [CrossRef]

[29] Chen, K., & Wang, J. (2023). Scheduling Hybrid Spark Jobs Based on Deep Reinforcement Learning. In *Proceedings of the 2nd International Conference on Big*

Data, Blockchain, and Economy Management (ICBBEM 2023). May 2023.

[30] Kiani, A., & Ansari, N. (2024). Hierarchical Adaptive Federated Reinforcement Learning for Efficient Resource Allocation in IoT Networks. *Computer Communications*, January 2024.

[31] Alipio, M., & Bures, M. (2023). Deep reinforcement learning perspectives on improving reliable transmissions in iot networks: Problem formulation, parameter choices, challenges, and future directions. *Internet of Things*, *23*, 100846. [CrossRef]

[32] Song, X., Liu, L., Fu, J., Zhang, X., Feng, J., & Pei, Q. (2023, December). A Reinforcement Learning-based DAG Tasks Scheduling in Edge-Cloud Collaboration Systems. In *GLOBECOM 2023-2023 IEEE Global Communications Conference* (pp. 1771-1776). IEEE. [CrossRef]

[33] Boateng, G. O., Erbad, A., Seid, A. M., Hamdi, M., Guo, X., & Guizani, M. (2024, December). Coalitional Game-guided Reinforcement Learning for P2P Resource Trading in Sliced IIoT Networks. In *GLOBECOM 2024-2024 IEEE Global Communications Conference* (pp. 644-649). IEEE. [CrossRef]

Dr. Meenakshi Garg is a dedicated academic professional currently serving as an Assistant Professor at Government Bikram College of Commerce, Patiala. With over 15 years of teaching experience, she has established herself as an expert in the fields of image processing and optimization. She holds an impressive academic portfolio, including a Ph.D., M.Tech., MCA, PGDCA, and is UGC NET qualified with an outstanding All India Rank (AIR) of 2.
Dr. Garg's contributions to academia and research are remarkable. She has been granted more than 10 patents and copyrights, alongside numerous research papers published in esteemed national and international journals. Her work reflects her deep commitment to innovation and excellence in her domain.
In addition to her technical expertise, Dr. Garg is known for her engaging teaching methods, inspiring students to excel in their studies and research endeavors. Her passion for knowledge dissemination and her ability to integrate theoretical and practical aspects make her a respected educator and mentor in her field. (Email: mouryagarg2005@gmail.com)