International Innovative Research Journal of Engineering and Technology
Vol. 8, Issue. 4, June 2023, pp. 01~16
ISSN: 2456-1983, DOI: 10.32595/iirjet.org/v8i4.2023.172

RPOM—Rational Process Offloading for Improving the Resource Utilization of the Internet of Things

S. S. Karthik[1], V. Vijayalakshmi[2], G. Zayaraz[3]
[1]SAP / ECE, Christ College of Engineering & Technology, Puducherry, India
[2]Associate Professor / ECE, Puducherry Technological University, Puducherry, India
[3]Professor/CSE, Puducherry Technological University, Puducherry, India

Article Info

Article history:

Received Mar 9, 2023
Revised May 10 2023
Accepted May 28, 2023

Keywords:

Decision-Making
IoT
Process Offloading
State Learning
Resource Utilization

ABSTRACT

The Internet of Things (IoT) interconnects diverse objects and service platforms for providing ubiquitous application support through diverse communication technologies. Quality of service and experience required application support is ensured through precise resource allocation and request scheduling in this platform. Considering the densely populated user scenario and service demand, this article introduces a Rational Process Offloading Method (RPOM) for reducing the service backlogs in IoT. This method distinguishes the scheduled and offloading of required application requests for preventing additional service delays. The decisions for offloading and time-sensitive application responses are performed using the state learning process. In this learning, the offloading and scheduling states are validated using current and previous state analysis for independent and congestion-less responses. The state is trained using the time and offloading demand observed between the request and response. However, the varying state modeling is used for determining a forecast-based service allocation, improving the utilization. The RPOM's performance is validated using the metrics of resource utilization, offloading ratio, service delay, and backlogs.

Corresponding Author:

S. S. Karthik,
SAP / ECE, Christ College of Engineering & Technology, Puducherry, India.
Email: karthsivashanthi@gmail.com

1. INTRODUCTION

Internet of Things (IoT) is a process that helps to connect physical objects or things based on embedded software, sensors, and technologies to provide better communication and data exchange processes for various fields. IoT is one of the rapidly evolving technologies which help to connect objects and things [1]. Service scheduling is an important process in every IoT network which helps to ensure both resource allocation and scheduling process for performing a particular task. Service scheduling uses an optimal scheduling algorithm which is used to utilize the CPU and helps to perform certain services [2]. Both latency rate and energy consumption rate are reduced with the appropriate service scheduling process in IoT. The service scheduling process improves the overall throughput and utilization of the CPU which helps to increase the performance and efficiency of IoT-based applications [3]. Service scheduling based on an edge-computing system is used in IoT-based applications, which helps to provide a proper set of scheduling processes for performing particular tasks. Load balancing strategy is used in the service scheduling process to increase the accuracy rate in the scheduling process. The service scheduling process improves both the accuracy and effectiveness of the system [4, 5].

Process offloading plays a vital role in the Internet of Things (IoT) which helps to provide an accurate set of resource allocation processes to perform or execute tasks in an application. The process offloading process

is also used to provide actual steps or ways to perform particular tasks in IoT-based applications [6]. A large-scale offloading approach is used in IoT to enhance the aspects and features which are needed for performing particular tasks. Auto scaler is used in the large-scale offloading approach which helps to identify the workloads and problems which are available while performing tasks [7]. Auto scalers provide an accurate set of data and allocation for performing tasks which helps to improve the overall performance and efficiency of IoT-based applications [8]. Computation offloading using an edge-computing system is also used in IoT-based applications to provide a significant set of processes and also eliminated unwanted problems which are occurred during processing data [9]. Dynamic computation offloading algorithm (DCOA) is used here for providing a better optimization process which helps to minimize the cost and latency rate in the computation process. DCOA provides possible solutions to solve problems to improve the efficiency and effectiveness of IoT-based applications. DCOA also reduces the workload and sub problems which provide a better accuracy rate in the detection process [3, 10].

Resource utilization is one of the processes which are available in the Internet of Things (IoT) which utilize resources to run particular composite services or tasks which are given by the users. The resource utilization process provides better interaction and communication services to the users improve the overall application performance and feasibility [11]. The proper resource allocation process plays a vital role in every IoT-based application to ensure the accuracy rate in providing services for the networks [12]. Machine learning (ML) techniques are mostly used in the resource utilization process in IoT systems provides an actual set of resources that are needed for performing tasks. ML techniques improve the effectiveness of the network which helps to enhance the performance and trustworthiness of the users [13]. The blockchain approach is used in the resource utilization process which helps to analyze every resource to perform a task in IoT. The blockchain approach is mostly used in IoT to provide a safe and secure communication process for the users which helps to enhance the feasibility of the network. ML techniques are mostly used to detect anomalies and problems which are available during the resource utilization process, which also provides a solution for the problems [14, 15].

2. RELATED WORKS

Aazam et al. [16] introduced a new task offloading method for the Internet of Things (IoT) named, the three-tier IoT-Fog-Cloud model using a cloud computing system. The proposed method is mostly used in wearable devices to increase the scalability and reliability of the system by providing better services for the users. IoT nodes play a vital role in the task execution process which also reduces both time and energy consumption rate in the computation process. Experimental results show that the proposed method improves the overall efficiency and performance of IoT-based applications and devices.

Misra et al. [17] proposed a new dynamic task offloading approach for Internet of Things (IoT) based applications using a software-defined fog computing system. The Task computation process is done based on optimal node selection, node usage, and paths improve the accuracy rate in the offloading process. When compared with other traditional methods, the proposed offloading approach reduces the latency rate and energy consumption rate in the computation process which helps to increase the overall efficiency and performance of the applications.

Adhikari et al. [18] introduced a delay-dependent and priority-aware task offloading scheme (DPTO) based on the fog computing framework for the Internet of Things (IoT). The proposed DPTO is used to form a multi-level sequenced queue to perform particular tasks based on priority and deadline aspects which helps to reduce the latency rate in waiting time. DPTO also reduces the optimal problems which are occurred while performing tasks in the offloading process. Experimental results show that the proposed DPTO scheme reduces the overall offloading time and helps to improve the reliability and efficiency of the system.

Hwang et al. [19] introduced a slicing and task offloading method for an edge computing system. First edge nodes are adjusted using a certain set of features to perform the virtualization process which helps enhance the effectiveness of the system. The proposed method is also used to reduce the functions which are needed for the optimization process. When compared with other methods, the proposed offloading methods improve the performance by reducing the latency rate and energy consumption rate in the task offloading process.

Liu et al. [20] proposed a parallel offloading of splittable tasks (POST) for heterogeneous fog networks. Generalized Nash equilibrium is used in the proposed POST approach to finding out the problems which are available in the computation process. When compared with other methods, the proposed POST model improves the overall performance and feasibility of the network by reducing the latency rate in the computation process.

Xiao et al. [21] proposed a task offloading and resource allocation algorithm (TO-RA) for an Internet of Things (IoT) based application. The blockchain approach is used in TO-RA to enhance the accuracy rate in the resource allocation process which helps to improve the accuracy rate in the allocation process. A virtual machine mapping strategy is used in the task offloading process which helps to allocate the accurate set of data for offloading process. Experimental results show that the proposed TO-RA method increases the efficiency and performance of the network.

Cui et al. [22] introduced a new task offloading method for Mobile edge computing (MEC) based on evolutionary stability strategy (ESS) in the Internet of Things (IoT) devices. The proposed method is mostly used to reduce both delay rate and waiting time for offloading process which helps to enhance the overall effectiveness of the network. ESS is used to perform a multi-level computation process which helps to reduce the energy consumption rate and helps to improve the efficiency of the network. Experimental results show that the proposed method improves the performance and reliability of IoT-based devices and applications.

Shahryari et al. [23] proposed a new task offloading scheme for fog-enabled Internet of Things (IoT) networks. The proposed method is mainly used to complete the task within the deadline and helps to reduce the delay rate in the task completion process. A sub-optimal algorithm is used in the proposed method to perform the swarm optimization process which helps to solve the optimization problems which are presented in the task offloading process. When compared with other traditional methods, the proposed scheme increases the overall performance by reducing the energy consumption rate in the computation process.

Hossain et al. [24] introduced a reinforcement learning-based task offloading approach for the Industrial Internet of Things (IIoT). The proposed approach provides a better resource allocation process which helps to reduce the energy consumption rate in the decision-making process. Experimental results show that the proposed approach reduces the cost and time consumption rate of the process which helps to increase the overall efficiency and effectiveness of the network.

Almutairi et al. [25] proposed a novel task offloading approach for the edge-cloud network. A fuzzy logic algorithm is used in the proposed method to perform the resource allocation process which helps to improve the overall feasibility of the network. The proposed method is mostly used to reduce latency-sensitive rates while performing tasks. When compared with other methods, the proposed improves the performance of the system by reducing the latency and time consumption rate in the computation process.

Mahini et al. [26] introduced a new evolutionary game theory approach for the task offloading process in a fog-computing network. The proposed method is used to reduce both time and energy consumption rates in the task computation process which helps to improve the increase the efficiency and feasibility of the network. Experimental results show that the proposed approach increases the overall coverage rate and accuracy rate in performing tasks which helps to enhance the trustworthiness of the network.

Sun et al. [27] introduced a task value-aware offloading scheme for the Internet of Things (IoT) based edge computing system. Both edge nodes (EN) and mobile equipment (ME) play a vital role in the task offloading process which providesan accurate set of information and resources for the computation process. Experimental results show that the proposed approach increases the effectiveness and scalability of the network.

Sun et al. [28] introduced a new resource allocation and task offloading scheme for fog-cloud architecture in the Internet of Things (IoT). The proposed method improves the efficiency of the network by providing an accurate task offloading process. When compared with other methods, the proposed task offloading approach reduces the energy and time consumption rate of the network which helps to improve the overall performance and effectiveness of the system.

Mu et al. [29] proposed a stochastic offloading control approach for Internet of Things (IoT) based applications and devices. Markov's decision-making process is used in the proposed method to allocate resources that are needed for the computation process and also helps to provide better cost-effective services to the users. Experimental results show that the proposed method reduces the energy consumption and time consumption rate of the process which helps to improve the efficiency and performance of the network.

3. PROPOSED METHOD

The proposed RPOM focusses on reducing service backlogs due to increasing user density and service demands. In particular, the time-sensitive application demands are serviced promptly by deciding over loading. The offloading process relies on service delays and current backlogs. Therefore, the proposed method operates

between the service provides and resources in handling user requests. An illustration of the proposed method is presented in Fig.1.

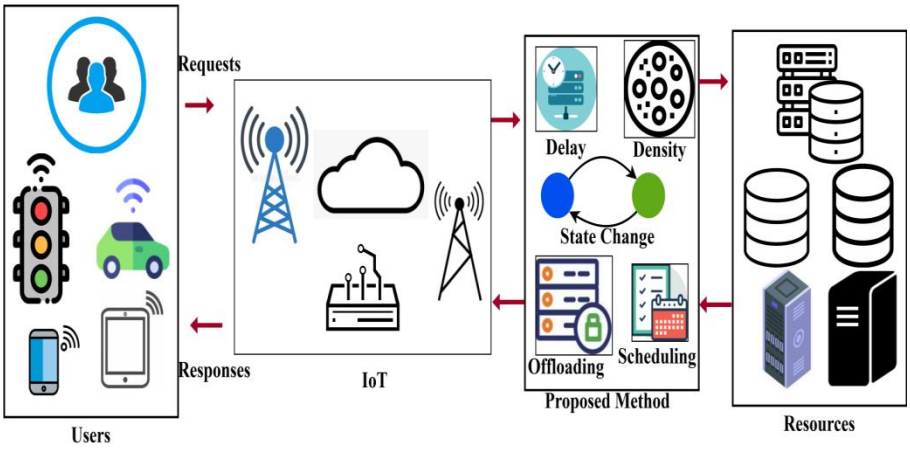

Figure 1. Proposed Method in IoT

The proposed method (Refer to Fig. 1) relies on state changes for deciding offloading/ scheduling. It changes with user density and request backlogs, and hence maximum resource utilization is observed. Based on different request-response slots, the scheduling is performed for resource allocation. In the following sections, the problem definition and RPOM briefings are presented.

Problem Definition: The proposed method observes the scheduling/ offloading as a linear problem wherein R requests receive S responses provided $R = S$. This linear problem is modified as an optimization concern if $R > S$ is observed. In this optimization, the backlogs B throughout the service interval is defined as

$$\left.\begin{array}{c} \sum_{i=1}^{t} B_i = \left(\frac{S_1}{R_1}dt + \frac{S_2}{R_2}dt + \cdots + \frac{S_i}{R_i}dt\right) = \sum_{i=1}^{t}\frac{S_i}{R_i}dt \\ such\ that \\ t.\frac{dS}{dt} = \int (R-S).\frac{(t_r - t_s)}{dt}; t(t_r - t_s) \end{array}\right\} \quad (1)$$

In equation (1), the variables $t, t_r,$ and t_s denote the interval, request, and response time. If the condition $R > S$ is achieved then t is extended for $(t_r - t_s)\ \forall\ (i+1)$ perspectives. This requires either offloading or scheduling for addressing $t \in (i+1)$ and increasing B issues. In this process, Q-process, Q-learning based state modeling is performed for improving resource utilization and response rate. This process is described in the following section.

State Modeling: The two definite processes namely offloading and scheduling are defined as two states in the learning process. Based on the response demand, and user density, the process is swapped between the states. In the state models, the decisions are made using a balancing factor (Δ). The high is the Δ representation for either state transitions, the process is performed. Based on R and user densities, the Δ varies and hence the optimization for equation (1) is achieved. The Δ factor is estimated for two optimization constraints as in equation (2). For $R > S$ and $t \in (i+1)$ constraints in the previous equation,

$$\Delta = \left\{\begin{array}{l} \frac{R}{S}.\left[1 - \frac{(t_r - t_s)}{t}\right], \forall\ B_i\ and\ i \in t \\ \frac{t_s}{t_r}.\left(1 - \frac{1}{R}\right) \forall\ t \in (i+1)(or)t+1 \end{array}\right\} \quad (2)$$

This Δ is estimated if $R > S$ (or)$(t_r - t_s)$is deficient in generating S. Therefore, either of the states is activated for maximizing responded. In particular, the state with high Δ is explored for its associated function. The state transition process is presented in Fig .2.

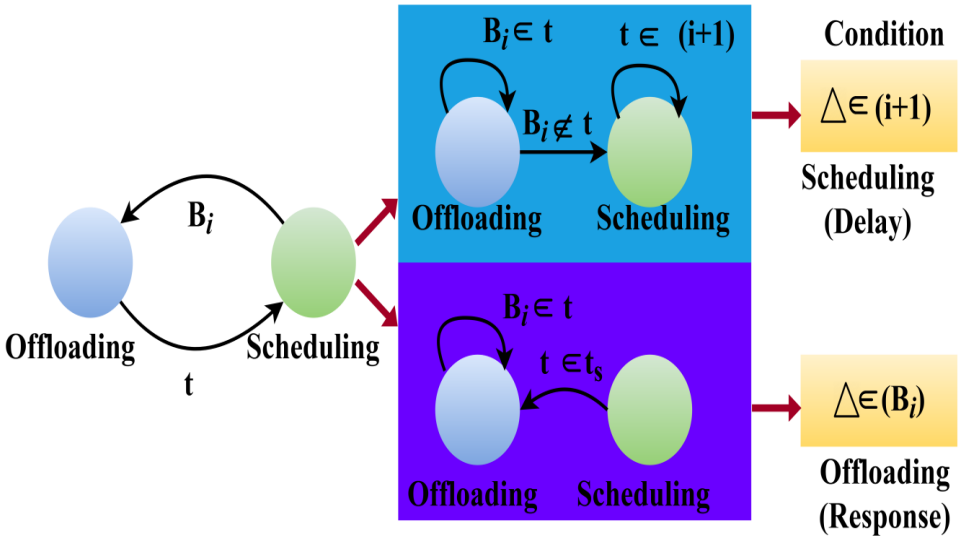

Figure 2. State Transition Process

The Δ estimation as in equation (2) determines its position in t or$(i+1)$ such that either of the processes are performed. The self-transition relies on$B_i \in t$ throughout S (or) $t \in (i+1)$ from the second S. Contrarily the transition state changes if Δ is classified under delay (i.e.) $(i+1)$ for scheduling (or) B_i for offloading. These two classifications address the delay and response optimizations satisfying equation (1). Post the every $i \in \left(\frac{S}{R}\right)$, the $\Delta$is estimated for both states. The high Δ in any of the state retains is transition for performing a task. This is rational preventing the existence of a single process in both states. Now, based on the Δ, the constraints in equation (1) are suppressed as a linear representation using equation (3)

$$\left.\begin{aligned} \forall\, B_i \in t, \frac{1}{R}dt = \Delta \int dt + \Delta \int S\, dt - \frac{1}{R-S}dt \\ \forall\, t \in (i+1), (t_r - t_s)dt = \frac{1}{\Delta}\int S dt - \int (R-S).\frac{dS}{dt} \end{aligned}\right\} \quad (3)$$

In equation the required (achievable) representations for B and t are presented using Δ such that if $\Delta <$ 1, then time slots varies requiring a scheduling. Contrarily, the $\frac{1}{R-S}$ is balanced to O such that it becomes in valid $\forall\, B$, requiring offloading. Based on the above representation, the analysis on offloading and scheduling are discussed in the following sections.

Offloading Process: The offloading process is required to reduce the backlogs in handling density unidentifiable user $R's$. Let N denote the set of resources for which R is assigned int_r. It generates S response att_s such that for an optimal interval $R = S$ is achieved in t. The optimization as represented in equation (1) and equation (3) requires precise state changes. This is validated for improving offloading requirements and hence backlogs are suppressed. Depending on the available t (allocated), the validations are provided in maximizing S for all R. The optimization (linear) is modified for the R that are processed outsite $'t'$. The offloading is performed within the N such that resource utilization is maximized. The utilization(R_u) before and after offloading is estimated using (4)

$$R_u = \left\{\begin{matrix} \arg\max\left(\frac{P}{N}\right)_i \;\; \forall\, i \in t, before\; offloadig \\ \left[\frac{P}{N} - \left(\Delta \times \frac{S}{R}\right)\right]_i \;\; \forall\, i \in (t+1), after\; offloading \end{matrix}\right\} \quad (4)$$

From equation (4), the Δ changes are performed for two distinct closes namely$B_i \in t$ and$B_i(t+1)$. Here, $(t+1)$ denotes the additional consecutive t in which B is reduced. Therefore, the state changes are modeled for t and $(t+1)$, derived from offloading. This is represented in Fig. 3.

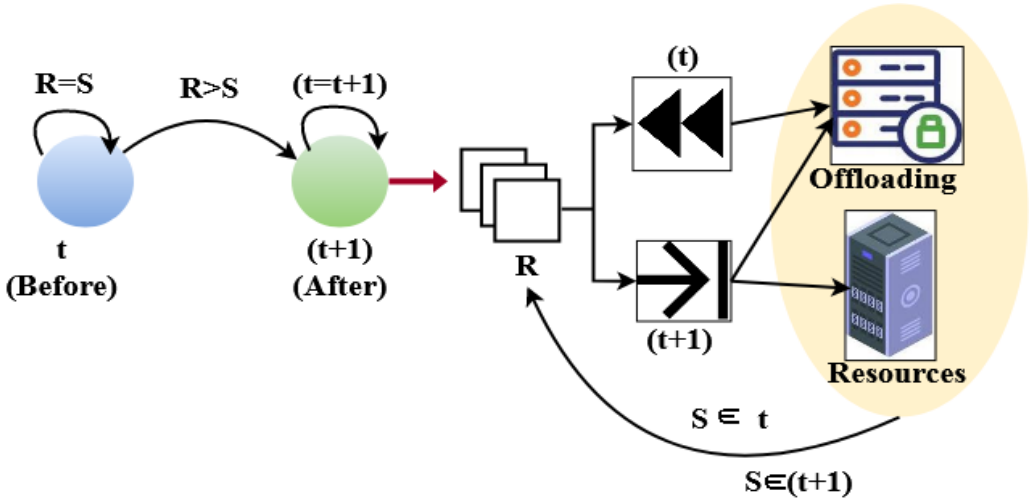

Figure 3. State Change for Offloading

The state changes between t and $(t+1)$ are modeled for R int_r that $S \in t$ and $S \in (t+1)$are reverted in $t = (t_r - t_s + 1)$. Therefore, the changes in Δ for t and$(t+1)$ is estimated as in equation (5).

$$\left.\begin{array}{l|l} \Delta_2 = \left(\frac{t_s}{t_r}\right)_1 - \left[\frac{(R-S)}{R}\right]_1 & \Delta_t = \left(\frac{t+1}{t_r}\right)_{t-1} - \left(\frac{S}{R}\right)_{t-1} \\ \Delta_3 = \left(\frac{t_s}{t_r}\right)_2 - \left[\frac{(R-S)}{R}\right]_2 & \Delta_{t+1} = \left(\frac{t+1}{t_r}\right)_t - \left(\frac{S}{R}\right)_t \\ \vdots & \vdots \\ \underbrace{\Delta_t = \left(\frac{t_s}{t_r}\right)_{t-1} - \left[\frac{(R-S)}{R}\right]_{t-1}}_{Before} & \underbrace{\Delta_{t+A} = \left(\frac{t+1}{t_r}\right)_{t+s-1} - \left(\frac{S}{R}\right)_{t+S-1}}_{After} \end{array}\right\} \quad (5)$$

The before and after Δ estimation is identified for$B_i \forall\, i \in t$ such that the offloading and non-offloading processes are defined. The before classification requires non-offloading where equation (3)'s linearity is retained. It is verified by balancing equations (2) $\forall\, B_i$ withΔ_t and Δ_{t+s} considering the offloading requirements.

$$\left.\begin{array}{l|l} \frac{R}{S}.\left[1 - \frac{(t_r - t_s)}{t}\right] = \frac{t_s}{t_r} - \frac{(R-S)}{R} & \frac{R}{S}\left[1 - \frac{(t_r - t_s)}{t}\right] = \frac{t+1}{t_r} - \frac{(R-S)}{R} \\ \frac{t_s}{t_r} = \frac{R-S}{R} & \frac{t+1}{t_r} = \frac{R-S}{R} \\ and\ Rt = St_r & and\ [1-(t+1)R = St_r \end{array}\right\} \quad (6)$$

The above estimated time requirements are used to suppress the backlogs due to offloading. The first estimation $Rt = St_r$ is required for optimal (linear) condition satisfaction as in equation (3). On the contrary, the extended $(t+1) \forall\, R$ achives backlogs less S maximizingR_u. This offloading pursued mitigates B in the joint intervals for maximizingR_u. For a linear offloading as per equation (3),Δ_1 toΔ_t achieve maximumR_u. Contrarily, if an offloading is required, thenΔ_{t-1} toΔ_{t+S-1} is the span for maximizingR_u.In the offloading process, as mentioned earlier, the rational allocation of $R \forall\, t$ and $(t+1)$ are achieved in coherent toR_u.

Scheduling Process: The scheduling is required for preventing additional time in providing S regardless of users and process. In the scheduling process, the second optimization in equation (1) is addressed. The linear coherence for equation (3) $\forall\, t \in (i+1)$ is analyzed for preventing additional delay. The B causing processes are suppressed by offloading P with the available N. This hinders the R allocation and hence the consecutive S requires optimalR_u maximization instances. The scheduling process is refined for $R \in (t+1)$ and does not satisfy equations (5) and (6). The conventional first-in-first-out and sliced allocations to N are exempted in this method. This is due to congested R and varying users that augments time (delay) for$(t+1)R$ and also R is classified under $(t+1)$ resulting in B. The scheduling is formulated between t and$(t+1)$ as given in equation (7)

$$\left.\begin{array}{c} \forall\, R\ \in t, \tau . N = R.P_i \,\forall\, i \in t \\ provided\ argmax\ \left\{\frac{P}{N_i}\right\}_i\ \forall\, i\ \in t_s \\ Contrarily, \\ \forall \left(\frac{R-S}{R}\right)\ \in (t+1), \frac{\tau}{P} = \frac{(R-S)}{R}.\frac{1}{N} \\ such\ that\ \int \tau \frac{(t+1)}{dt} = \int \left(\tau - \frac{S}{R}\right)\frac{t}{dt} + \int \frac{S}{R}\frac{ds}{dt} \end{array}\right\} \quad (7)$$

In equation (7), τ denotes the N capacity for which the P is allocated. This P is either a complete R or a part of R that is balanced in time t to $(t+1)$,. The aim is to suppress the prolonging of P or R to $(t+1)$ and reducing delay. The state transition is modeled for P and R based on$(t+1)$ for preventing overflows and hence precise scheduling allocations are performed. The state transitions are illustrated in Fig. 4.

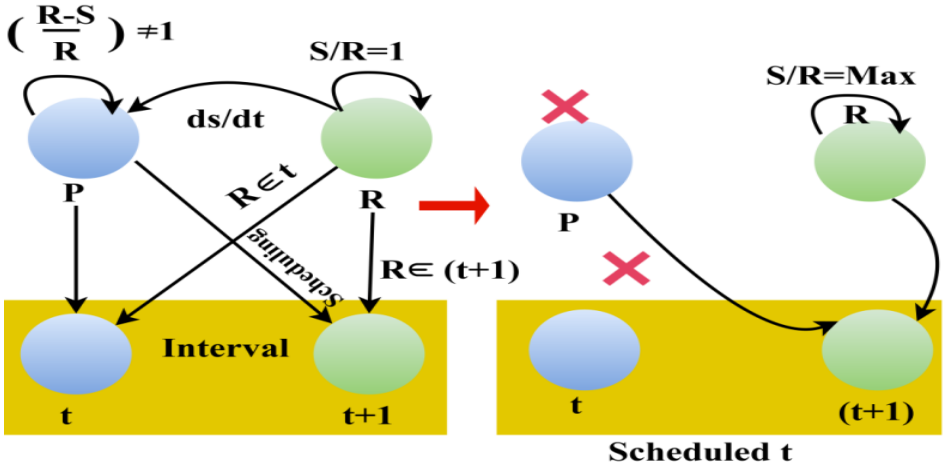

Figure 4. State Changes for P and R

The above representation filters t and$(t+1)$ for P and R such that the scheduling is performed in coherence to$\int \tau \frac{(t+1)}{dt}$. This scheduling process between t and$(t+1)$again relies on Δ for its occurrence between different allocations. The Δ is estimated for t to$(t+1)$ in a linear form such that$\sum t+1$ experiences the state changes using Δ. This is computed using equation (8) as

$$\Delta = \left\{\begin{array}{c} \frac{P}{R}\ \forall\, t, such\ that\ \tau . N = S \\ \left(\frac{R-S}{R}\right) + \frac{P}{S}, \forall\, t+1, such\ that\ \tau . \left(\frac{N}{P}\right) = S \end{array}\right\} \quad (8)$$

Based on this validation, the swapping of R and P is performed. In contrast to the validation process, the proposed method identifies available $(t+1)$ for P or R allocations. The allocations are non-linear in assigning N based on τ. Therefore, the R offloaded in $(t+1)$ are split into P based on τ, the B is suppressed by preventing N overflows. This is achieved by swapping P and R withint_s such that Δ [as in equation (8)] ensures the optimization in equation (1) using the scheduling between t and$(t+1)$. This confines the delay between overloaded and offloading requests int_r. An analysis on offloading ratio, t and $(t+1)$ requirements for the varying capacity is presented in Fig. 5.

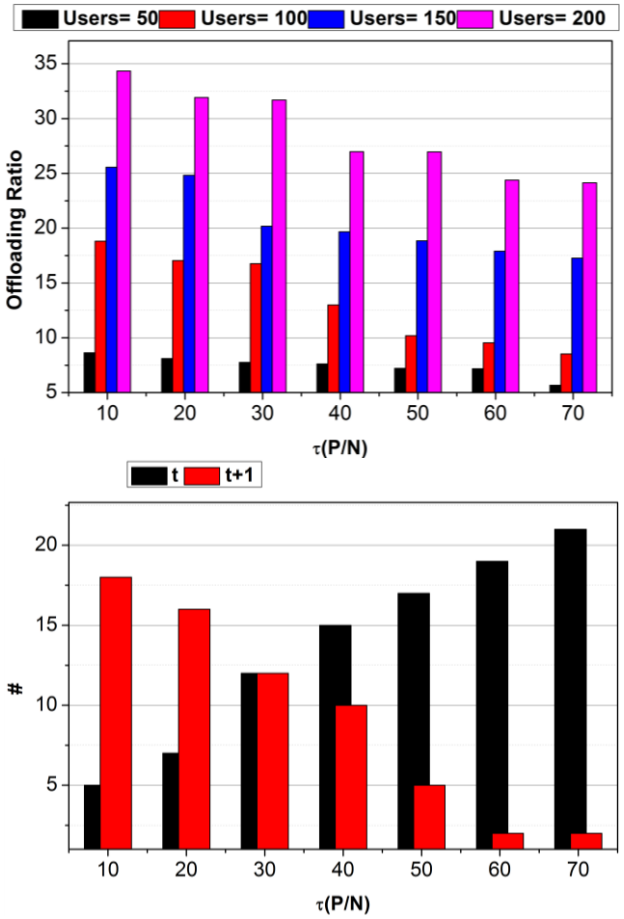

Figure 5. Offloading and $t, (t+1)$ Analysis for Capacity

In Fig.5, the analysis for offloading ratio and $t, (t+1)$ instances for varying τ is presented. The proposed method improves the t allocation by Δ in distinct intervals. This maximizes the allocations based on time and linear decisions. The constraints in equation (1) and its linear derivative equation (3) are suppressed using Δ appropriations. This confines the offloading ratio for varying t as in $(\sum t+1)$. As the offloading ratio decreases, $(t+1)$ is reduced wherein the need for P extraction from R is less. Therefore, based on Δ balancing as in equation (6), $\left(\frac{R-S}{R}\right)$ availability is confined $\forall(t+1)$. In this case, the further allocations are prevented in$(t+1)$, maximizing response. Therefore, for any user density, the shared $N's$ based on τ reduces$(t+1)$ intervals and hence the B is reduced. In Fig. 6, the B under varying swapping instances is analyzed.

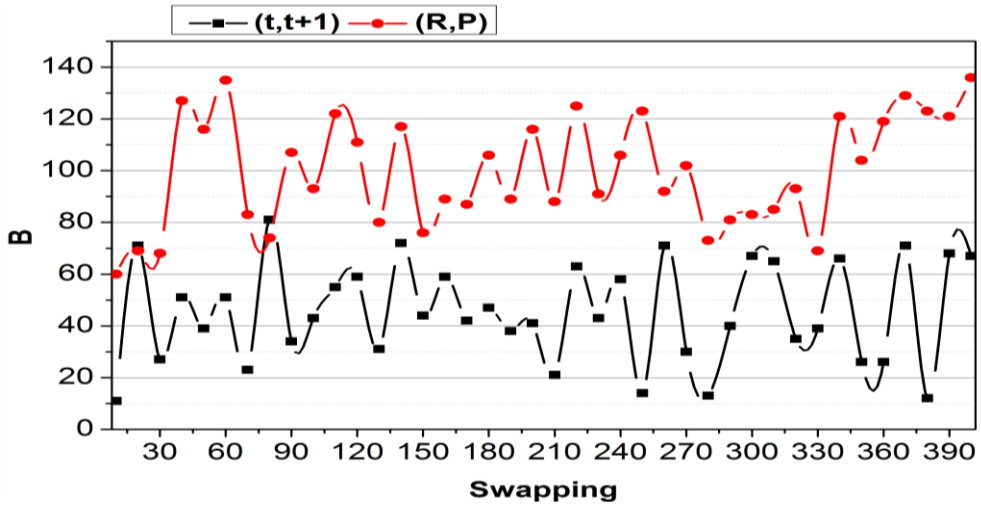

Figure 6. B for varying Swapping Instances

For the varying swapping counts of $(t, t+1)$ and (R, P), the B is analyzed in Fig.6. The state transitions of $[t, (t+1)]$ follows the linear optimization as before and after (i.e.) Δ_t and Δ_{t+s}. This is required for preventing B in $(t+1)$; the allocation follows $\left(\tau - \frac{S}{R}\right)\frac{t}{dt}$ and $\frac{(R-S)}{R}.\frac{1}{N}$ for precise sharing. In this, the allocations for scheduling reduces the B. Contrarily, the Δ as in equation (3) defines the swapping requirements for P and R such that $\frac{P}{S}$ and $\frac{P}{R}$ are the defining factors for maximizing responses. This is considerable based on the state swapping performed in $\int \frac{\tau(t+1)}{dt}$, preventing B.

4. DISCUSSION

The proposed method's performance is analyzed using the ContikiCooja simulations and discussed using a comparative analysis. The network with 220 users, 10 resources, and 8 access points is considered for analysis. The state transitions are varied between 20 and 400 iterations for identifying the minimum B. A total of 1100 request processes are handled in this experiment and the maximum time for backlog is 2.4s.Using this environment, the metrics resource utilization, offloading ratio, service delay, backlogs, and response ratio are analyzed by varying the users and processes. For a comparative analysis, the methods Detour [17], EdgeABC [21], and TOA-LS [25] are considered from the related works section.

Resource Utilization Analysis

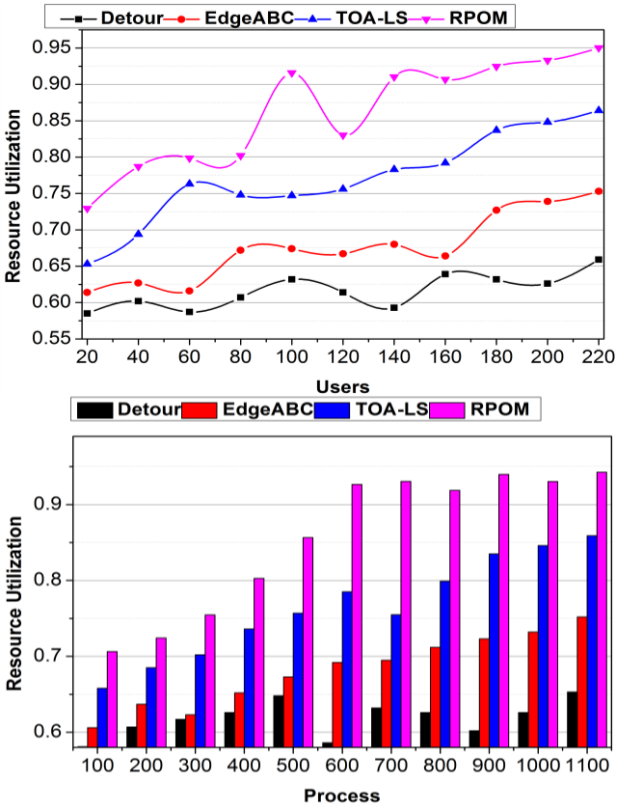

Figure 7. Resource Utilization

Fig. 7 presents the comparative analysis for resource utilization under varying users and processes. The proposed method maximizes R_u for P, R and $(t+1)$ such that Δ is validated. In the state classification process Δ is balanced from equation (2) for $(t+1)$ and P as defined in equations (5) and (8). First, the R_u is identified based on t and $(t+1)$ is estimated based on the swapping required. The Δ is balanced based on $\left(\frac{t_s}{t_r}\right)$ and St_r for improving the resource utilizations. The Δ defined cases for $[t,(t+1)]$ and (R,P) are optimal for allocating P and R among the available N. Therefore, the balanced allocations are swapped between the weighted $\Delta \forall \frac{\tau}{P}$ and $\frac{S}{R}$ such that $\frac{S}{R} = max$ is achieved. On the contrary the $R \in (t+1) \forall P$ is segregated and provided under distinguishable intervals. This maximizes the N utilization in t and $(t+1)$ for both P and R. Therefore, the consecutive states are optimized, improving the further resource utilization satisfying S.

Offloading Ratio Analysis

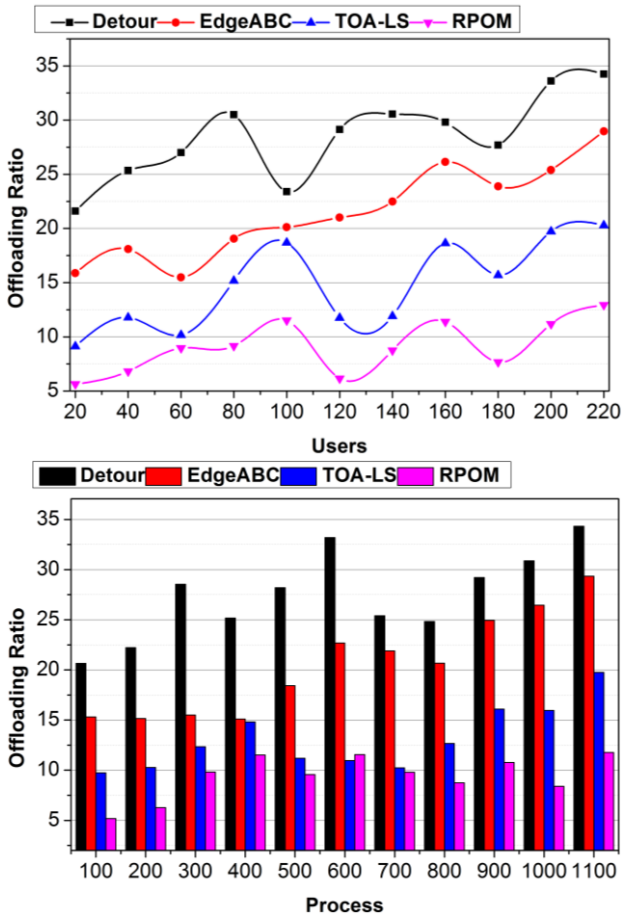

Figure 8. Offloading Ratio

The comparative analysis for offloading over varying users and processes is presented in Fig. 8. The proposed method reduces offloading ratio by splitting t and$(t+1)$ requirements based on state recommendations. The state recommendations are performed in two instances viz $t,(t+1)$ and $(t+s)$ swapping. This retains the availability of $N\forall R$ and hence the offloading requirements from $(t+1)$ to$(t+s)$ is less. In case of the P and R swapping, the R is split into $P \in (t+1)$ that is confined before $(t+s)$. However the change in swapping increases the one-point variations increasing the offloading demands. In this case, the chances are abrupt for further scheduling and hence the changes are optimized without further swapping. The proposed method maximizes the allocations with the linear optimization, suppressing the previous$\left(\tau - \frac{S}{R}\right)$. Therefore, the offloading from the one-point is decreased by allocating P (or) R within t and thereforeR_u is maximized.

Service Delay Analysis

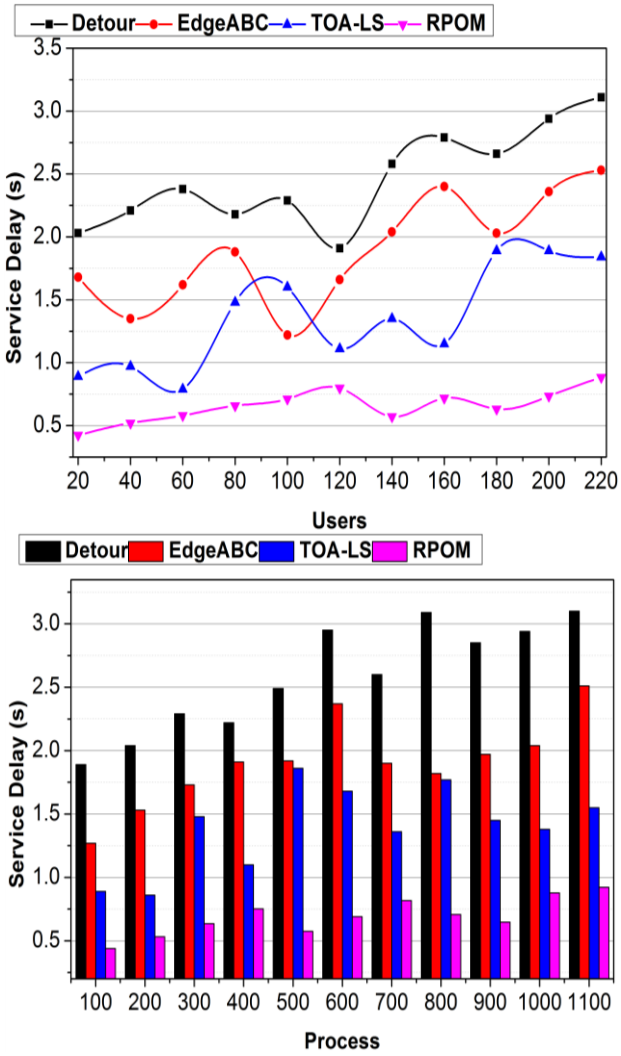

Figure 9. Service Delay

For the varying users and processes, the service delay is comparatively analyzed in Fig.9. This proposed RPOM achieves less delay compared to the other methods. The service delay due to $(t+1)$ to$(t+5)$ offloading generally increases the wait time of $R \in t_r$. In this method, the last offloading is confined based on the available N and its τ. This feature is pursued for R segregating P and hence further wait times are confined. By retaining the allocations in distinguishable intervals the proposed method achieves high allocations in t and $(t+1)$. This reduces the service delay for varying user densities improving the allocations in t. Further, P and R based classifications restrict the B due to $(t+1)$ to$(t+S)$ and hence$(t+1)$ is valid. If a uneven/ non-linear requirement is processed, then the variations are addressed from the consecutive$(t+1)$. Therefore, a concurrent allocation and S delivery is performed wherein$(t_r - t_s)$ is shared between R requiring less delay.

Backlogs Analysis

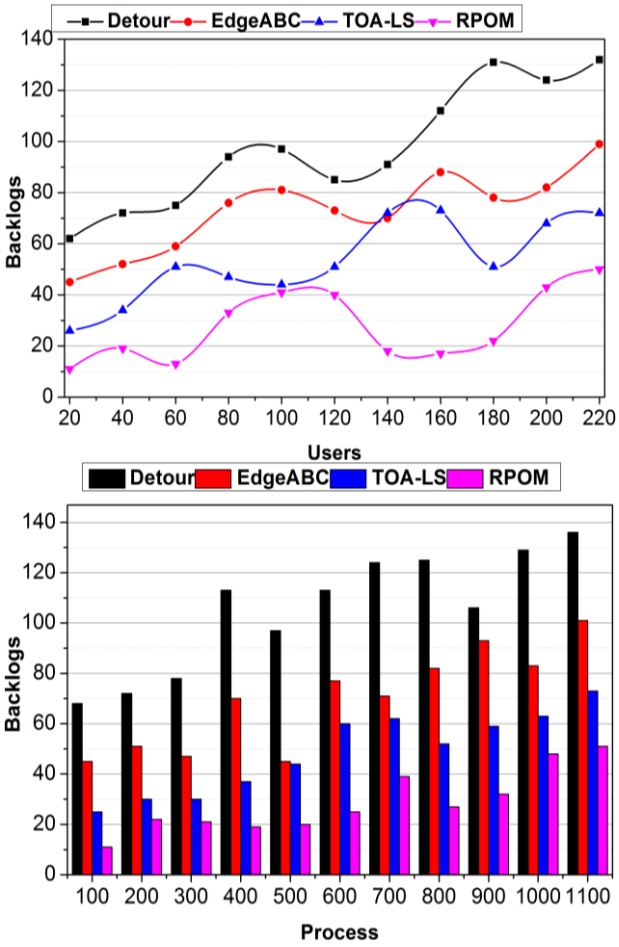

Figure 10. Backlogs

The proposed method achieves fewer backlogs compared to the other methods. The problem formulation is performed for $\frac{ds}{dt}$ under offloading conditions such that $(t_r - t_s)$ is limited. In the optimization process, the non-linear deviations are confined depending on $Rt = St_r$ verification. Based on the allocated instances, the B is confined in t. However, the above condition is not the same for distinguishable interval wherein $(R - S)\frac{ds}{dt}$ is required. These requirements are handling using diverse state analysis and Δ balancing for allocation. Therefore, the R (or) $P \in (t + 1)$ and beyond are suppressed using $\int \frac{S}{R}\frac{dS}{dt}$ in the scheduling process. The B that increasing in any $(t_r - t_s)$ interval is confined by swapping allocations based on state decisions. Therefore, the Δ_t and Δ_{t+s} are the required balancing factors user different allocations for improving responses and confining B. The confinement is monotonous for varying users and process achieving fewer backlogs as presented in Fig. 10.

Response Ratio Analysis

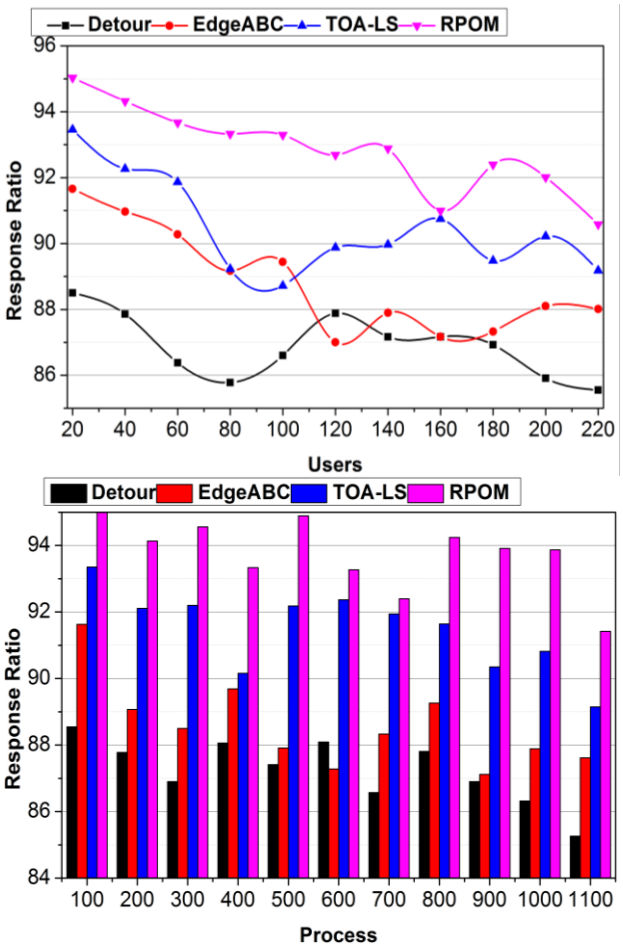

Figure 11. Response Ratio

The comparative analysis for response ratio over the varying users and process is presented in Fig. 11. The proposed method maximizes response based on the B minimization and precise N allocation. The constructive features such as R_u maximization, offloading minimization and delay minimization favors S. In the state transitions, the available constraints post linear rectifications are exploited for improving S. The splitting conditions for $\left(\tau - \frac{S}{R}\right)$ and $(R - S)\frac{ds}{dt}$ for scheduling and offloading in the respective orders are handled using Δ. The proposed method maximizes $S\,\forall t$ and the remaining is handling in $(t+1)$ interval. In the contrary case, if the user and P density increases, then a slight deviation is observed. Based on the available N and further segregated P, B is confined between $(t+1)$ and $(t+s)\forall\,\Delta$. Therefore, the allocated intervals respond to R, maximizing S. In Table 1, the comparative analysis results for varying users and process is presented.

Table 1. Comparative Analysis Results

Metrics	Detour	EdgeABC	TOA-LS	RPOM
Varying Users				
Resource Utilization	0.659	0.753	0.864	0.9501
Offloading Ratio	34.25	28.97	20.3	12.946
Service Delay (s)	3.11	2.53	1.84	0.884
Backlogs	132	99	72	50
Response Ratio	85.55	88.01	89.18	90.584
Varying Process				
Resource Utilization	0.653	0.752	0.859	0.9427

Offloading Ratio	34.32	29.35	19.75	11.765
Service Delay (s)	3.1	2.51	1.55	0.923
Backlogs	136	101	73	51
Response Ratio	85.26	87.62	89.15	91.418

5. CONCLUSION

For addressing the backlog issues in time-sensitive Internet of Things applications, this article introduced the rational process offloading method. This method differentiates scheduling and offloading intervals for the varying processes and requests. In this differentiation, state learning is exploited for balancing the request allocations. The state learning operates between time intervals and processes for preventing backlogs in the allocated intervals. Besides, the linear optimization-based request processing and scheduling increases the resource utilization for varying user densities. The current and consecutive states are explored for improving the response rate in the different intervals. Based on the linear constraint satisfaction and request allocation with the capacity considerations, the proposed method maximizes response ratio. For the varying users, the proposed method achieves 9.57% high resource utilization, 14.89% less offloading ratio, 10.77% less service delay, 12.67% less backlogs, and 9.01% high response ratio.

REFERENCES

[1] Alameddine, H. A., Sharafeddine, S., Sebbah, S., Ayoubi, S., &Assi, C. (2019). Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. *IEEE Journal on Selected Areas in Communications*, *37*(3), 668-682.

[2] Hazra, A., Adhikari, M., Amgoth, T., &Srirama, S. N. (2020). Joint computation offloading and scheduling optimization of iot applications in fog networks.*IEEE Transactions on Network Science and Engineering*, *7*(4), 3266-3278.

[3] Kim, H. W., Park, J. H., &Jeong, Y. S. (2019). Adaptive job allocation scheduler based on usage pattern for computing offloading of IoT. *Future Generation Computer Systems*, *98*, 18-24.

[4] Azizi, S., Shojafar, M., Abawajy, J., &Buyya, R. (2022). Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach. *Journal of Network and Computer Applications*, 103333.

[5] Siar, H., &Izadi, M. (2021). Offloading Coalition Formation for Scheduling Scientific Workflow Ensembles in Fog Environments.*Journal of Grid Computing*, *19*(3), 1-20.

[6] Kim, Y., Song, C., Han, H., Jung, H., & Kang, S. (2020). Collaborative task scheduling for IoT-assisted edge computing.*IEEE Access*, *8*, 216593-216606.

[7] Hussein, M. K., &Mousa, M. H. (2020). Efficient task offloading for IoT-based applications in fog computing using ant colony optimization.*IEEE Access*, *8*, 37191-37201.

[8] Qi, H., Mu, X., & Shi, Y. (2020). A task unloading strategy of IoT devices using deep reinforcement learning based on mobile cloud computing environment. *Wireless Networks*, 1-11.

[9] Jin, W., Lim, S., Woo, S., Park, C., & Kim, D. (2022). Decision-making of IoT device operation based on intelligent-task offloading for improving environmental optimization.*Complex & Intelligent Systems*, 1-20.

[10] Nwogbaga, N. E., Latip, R., Affendey, L. S., &Rahiman, A. R. A. (2021). Investigation into the effect of data reduction in offloadable task for distributed IoT-fog-cloud computing. *Journal of Cloud Computing*, *10*(1), 1-12.

[11] Qu, G., Wu, H., Li, R., & Jiao, P. (2021). Dmro: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing. *IEEE Transactions on Network and Service Management*, *18*(3), 3448-3459.

[12] Xu, S., Liu, Q., Gong, B., Qi, F., Guo, S., Qiu, X., & Yang, C. (2020). RJCC: Reinforcement-learning-based joint communicational-and-computational resource allocation mechanism for smart city IoT. *IEEE Internet of Things Journal*, *7*(9), 8059-8076.

[13] Alfarraj, O. (2021). A machine learning-assisted data aggregation and offloading system for cloud–IoT communication.*Peer-to-Peer Networking and Applications*, *14*(4), 2554-2564.

[14] Mirmohseni, S. M., Tang, C., &Javadpour, A. (2020). Using markov learning utilization model for resource allocation in cloud of thing network.*Wireless Personal Communications*, *115*(1), 653-677.

[15] Jaddoa, A., Sakellari, G., Panaousis, E., Loukas, G., &Sarigiannidis, P. G. (2020). Dynamic decision support for resource offloading in heterogeneous Internet of Things environments.*Simulation Modelling Practice and Theory*, *101*, 102019.

[16] Aazam, M., Islam, S. U., Lone, S. T., & Abbas, A. (2020). Cloud of things (CoT): cloud-Fog-IoT task offloading for sustainable internet of things. *IEEE Transactions on Sustainable Computing*.

[17] Misra, S., &Saha, N. (2019). Detour: Dynamic task offloading in software-defined fog for IoT applications. *IEEE Journal on Selected Areas in Communications*, *37*(5), 1159-1166.

[18] Adhikari, M., Mukherjee, M., &Srirama, S. N. (2019). DPTO: A deadline and priority-aware task offloading in fog computing framework leveraging multilevel feedback queueing. *IEEE Internet of Things Journal*, *7*(7), 5773-5782.

[19] Hwang, J., Nkenyereye, L., Sung, N., Kim, J., & Song, J. (2021). IoT service slicing and task offloading for edge computing.*IEEE Internet of Things Journal*, *8*(14), 11526-11547.
[20] Liu, Z., Yang, Y., Wang, K., Shao, Z., & Zhang, J. (2020). Post: Parallel offloading of splittable tasks in heterogeneous fog networks. *IEEE Internet of Things Journal*, *7*(4), 3170-3183.
[21] Xiao, K., Gao, Z., Shi, W., Qiu, X., Yang, Y., &Rui, L. (2020). EdgeABC: An architecture for task offloading and resource allocation in the Internet of Things. *Future Generation Computer Systems*, *107*, 498-508.
[22] Cui, Y., Zhang, D., Zhang, T., Chen, L., Piao, M., & Zhu, H. (2020). Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices. *AEU-International Journal of Electronics and Communications*, *118*, 153134.
[23] Shahryari, O. K., Pedram, H., Khajehvand, V., &TakhtFooladi, M. D. (2021). Energy and task completion time trade-off for task offloading in fog-enabled IoT networks.*Pervasive and Mobile Computing*, *74*, 101395.
[24] Hossain, M. S., Nwakanma, C. I., Lee, J. M., & Kim, D. S. (2020). Edge computational task offloading scheme using reinforcement learning for IIoT scenario.*ICT Express*, *6*(4), 291-299.
[25] Almutairi, J., &Aldossary, M. (2021). A novel approach for IoT tasks offloading in edge-cloud environments.*Journal of Cloud Computing*, *10*(1), 1-19.
[26] Mahini, H., Rahmani, A. M., &Mousavirad, S. M. (2021). An evolutionary game approach to IoT task offloading in fog-cloud computing. *The Journal of Supercomputing*, *77*(6), 5398-5425.
[27] Sun, J., Wang, H., Feng, G., Lv, H., Liu, J., &Gao, Z. (2022). TOS-LRPLM: a task value-aware offloading scheme in IoT edge computing system. *Cluster Computing*, 1-17.
[28] Sun, H., Yu, H., Fan, G., & Chen, L. (2020). Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture. *Peer-to-Peer Networking and Applications*, *13*(2), 548-563.
[29] Mu, S., &Zhong, Z. (2020). Computation offloading to edge cloud and dynamically resource-sharing collaborators in Internet of Things.*EURASIP Journal on Wireless Communications and Networking*, *2020*(1), 1-21.