Date available online: 23-07-2025
Vol. 3, Issue 1 (January – June 2025)
This is an open-access article.
DOI: https://doi.org/10.24312/ucp-jeit.03.01.543

Smart Resource Allocation for Mobile Edge Network in IoT Using Game Theory

Samra Shereen[1], Asif Kabir[1], Syed Mushhad M. Gilani[2], Abdur Rehman Riaz[3], Zahid Mahmood[1]
[1]Department of CS and IT, University of Kotli, Kotli, Azad Jammu and Kashmir, Pakistan
[2]Department of Computer Science, Agriculture University Faisalabad, Pakistan
[3]Department of Computer Science, University of Management and Technology, Sialkot, Pakistan

Corresponding Author: Syed Mushhad M. Gilani (Email:mushhad@uaf.edu.pk)

ABSTRACT-

Emerging from years of research and development, the modern era of computing recognizes the Internet of Things (IoT) as the most empowering technology to connect the digital and real world. IoT has introduced new advancements that are transforming the world, however, it still faces constraints that limit its effectiveness in various application areas, including computing power, resource allocation, reliability, and time consumption. Achieving acceptable latency for task operations on IoT devices necessitates the appropriate allocation of Mobile Edge Caching device computing resources which should be based on task size, delivery, and service latency. It is impossible to handle the billions of data requests originating from a growing number of base stations. This research proposes a mechanism for allocating computing resources and caching to facilitate efficient scheduling in cellular networks. A game theory approach used to model miniaturization problems has been employed in this work. A wireless network system has been analyzed where each node in the system is a participant with its strategies and contributions to achieve the desired performance. The simulation results show that the proposed technique has great potential to improve resource allocation. Each IoT device increases the number of requests handled by the Mobile Edge Computing(MEC) server in the non-cooperative subgame. The proposed system efficiently allocates IoT resources excels in performance and reduces latency.

Index Terms: Internet of Things, Game Theory, Resource Allocation, Mobile Edge Computing

I. INTRODUCTION

Mobile networks have experienced rapid growth over the past decade, offering multimedia, online gaming, and video services. The number of mobile phone users and data traffic has exponentially grown [1]. IoT is one of the most emerging research driven by the widespread adoption of smart technology devices and advancement in communication technologies, including 5G. Wireless networks are expected to have an abundance of devices such as smartphones, portable computing devices, smart sensors, and other growing numbers of physical devices. These devices spread a large volume of data in the network, that's why networks need high-performance computing and a big storage capacity to manage it [2].

However, despite the seemingly unlimited computational capabilities offered by cloud services, this paradigm introduces several challenges like trust, congestion issues, high transmission costs, and prolonged service latency hinder its feasibility in many IoT scenarios that require real-time interaction or mobility. Mobile Edge Computing (MEC) solved the problem of heavy traffic in the network. At the edges of the network processing and caching of data are done. MEC consists of a single-edge server or group of devices that work together to serve mobile users. The MEC is a more

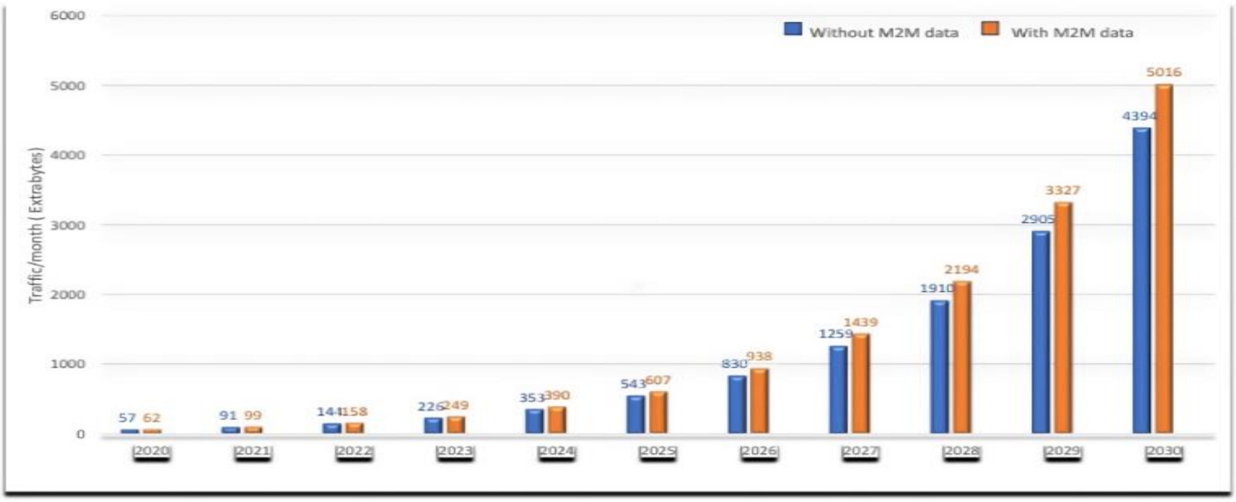

FIGURE 1: Global mobile devices and connection

efficient viable option than the remote cloud because it has much lower network latency. In addition, the MEC will be able to efficiently explore the computing and storage resources available at the edge of the network [3]. Data traffic generated by mobile devices all over the globe is forecast by ITU as shown in Figure 1. According to the studies data traffic rose up to 55% in the decade of 2020 to 2030. The estimated amount of data in 2025 will be 607 Exabytes and 5016 Exabytes in 2030(Source: Cisco).

However, MEC also has some limitations, such as the high cost of implementing and maintaining the architecture, and the enormous pressure that a complex and dynamic IT environment puts on MEC vendors. The rational allocation of computing and network resources to meet the polarization needs of mobile communications under dynamic MEC conditions is extremely challenging today [4]. In connection with the IoT, a neural network search system based on compound learning has been developed. It includes an anomaly classification model at the edge and a distributed learning framework for combining model parameters at the server to create a generic model for all edge regions. This method not only reduces data transmission requirements and increases latency, but also improves user privacy. Although the efficiency was slightly reduced due to the reduced location of the edge, accuracy was achieved by creating models suitable for different scenarios [5].

MEC servers also efficiently plan the resources of the mobile edge networks by joint caching and computing allocation mechanisms. In different environments, base stations (BSs) participate to control the computing space that can inhabit the MEC server to enhance the quality of use of their consumers [6]. Under the condition of mutant MEC, the experimental results show that the proposed Dynamic Reinforcement Learning Resource Allocation (DRLRA) algorithm performs better than the conventional algorithm. 5G wireless networks have been attracting much attention from academia and industry since the last quarter [7]. The biggest challenge is to meet the cost and energy consumption data compared to today's networks. 5G wireless cellular networks will borrow many new technologies to support the growth of wireless transmission services that are yet to be invented [8].
The network segment resource request mediation process improves the instantiation, configuration, and scaling of network segment resource requests when the client-provider relationship of 5G segments is broken. Likewise, the IoT industry can be optimally restructured to accommodate unexpected 5G network traffic. Interestingly, IoT Broker offers different functionalities.

(i) Appropriate selection of IoT Gateway (GW) configured to satisfy downstream order data request or Quality of Services (QoS) parameters (e.g., data accuracy, notification rate).
(ii) Measurement of data activity to cover business fluctuations, notification rate changes of IOT, and changes in QoS parameters during request/subscription delivery.
(iii) Data trading optimization to maximize the efficiency of 5G network applications [9].

Optimizing the allocation of their computing and communication resources, a protocol based on four specialized domains, and developing an energy-efficient design framework to meet the computational silence needs while reducing their overall consumption of energy. The IoT needs to be properly managed, and network performance needs to be improved. In a two-subcaste diverse IoT network with limited network coffers, a distributed Q-learning supplementary power distribution algorithm ensures the fairness of different biases to ensure druggies are treated equally [10].

In recent years, MEC, an efficient computing paradigm, has provided abundant computing resources for IoT. Overall, deploying MEC servers closer to mobile users effectively reduces access latency and the cost of using cloud services. Several mobile applications have been developed to connect the world of things to the Internet. However, to guarantee fair task action latency among IoT devices, calculation resources of MEC units need to be allocated accordingly based on task size whilst considering transmission and service latency. Using deep joint caching and computing learning algorithms, the goal of this study is to make it possible for manipulators to acquire new and challenging skills to solve the issue of resource allocation. How to reasonably allocate computing resources and network resources to meet the needs of mobile devices under the ever-changing conditions of MEC has become an important issue today. To address this problem, we propose an intelligent deep policy based on asset storage and processing learning, which can adaptively identify logs and network assets, reduce typical overtime, and balance the utilization of resources in different MEC conditions.

We yield algorithms from game theory and drive an efficient formula for smart resource allocation. The efficiency and accuracy of our driven formula are validated by using MATLAB. The tool gives us graphs that can show the accuracy of our proposed work with the growth of mobile biases and the improvement of communication skills and cognition, complex, multifaceted, and computationally intensive mobile processors have emerged. Due to limited resources, mobile polarization is increasingly limited. The research article is arranged as follows: Section II discusses the state of art technique from the literature. Our proposed model to describe the system is in Section III. In Section IV we articulate the problems related to the topic. Analysis and discussion of the proposed model is reframed in Section V and the conclusion of our studies is written in Section VI.

II.RELATED WORKS

Resource management algorithms can be distinguished based on their approach to resource allocation such as; Provision is the act of assigning resources to workloads. Allocation is the distribution of resources linking competing loads. Modeling provides a framework that assists in predicting the resources needed for a given workload [11]. Brokering is the negotiation of resources through an agent. Scheduling is organizing resources, requests, and events in a timetable that links requirements and time intervals for available resources [7]. Resource allocation in mobile edge computing using game theory faces challenges involving different technologies. However, its ability to cover different decision strategies helps to improve the decision process for the allocated time and objectives [8].

A game-theoretical approach to solving the attribution problem of IoT problems. There are three reasons for adopting a game-theoretic approach in this way. First of all, users of your application may have different needs and interests [12]. Game theory has been used successfully in many fields as an effective tool for analyzing the mutual influence of multiple actors acting on their interests. No application user has an incentive to unilaterally deviate, as an incentive-enabled mechanism can be jointly developed in an edge computing environment and is a satisfactory IoT solution. Second, Game seeks to harness the intelligence of individual application users to solve IoT problems in a decentralized way [13], [14]. This can reduce the high search load for centralized optimal solutions. The number of users assigned to the application and the number of edge servers available. Finally, comparing centralized and distributed game-theoretical approaches can quickly find an operating System solution. This allows applications to meet the needs of users and application providers for a low-latency edge computing environment [15]. Additionally, gaming application providers must consider capacity constraints such as CPU, memory, and bandwidth**.** etc. Compared to mega cloud servers data processing is limited in data centers, and edge server's capabilities are typically shared between multiple application vendors. So, the edge server must have enough computing capacity for application users to this edge server [16].

MEC is a promising way to expand the computing competencies of Western Digital (WD). MEC works silently by offloading some or all of the WDs' computing tasks to nearby MEC server access points [17]. Through MEC, small and low-power WDs can offload their computing tasks to access points, and those tasks can always be loaded and computed by embedded MEC servers. However, once the computational tasks are successfully offloaded, the MEC mode can facilitate computationally intensive tasks in real-time through both the original offset computing and the edge computing of the MEC service offloading tasks [18]. All of these are connected to the Internet and produce the low-speed tracking, dimensional, or robotic data that many businesses and end-users routinely require, underscoring the need for online coverage techniques in IoT enterprises. On the other hand, the number of devices on the Internet has recently added a new network factor the future of connectivity to "everything" on the Internet. IoT "Big Data" focuses on the four V's: Velocity, Variability, Volume, and Values. Then we have different data models, produced at different rates, which affect the different volumes of data that are dumped and used in IoT operations. Therefore, it is necessary to consider the latest technologies when handling data. The amount of data generated by mobile and IoT bias has increased significantly. These devices, such as smartphones, wearables, and detectors, have limited computing and energy resources. The decomposition process and inventory of resource-limited bias toward income currently face similar limitations [19].

However, computers are hosted in huge data centers located far from the extreme endpoints. In addition, the increased amount of changed data places a significant burden on network connections. Network functions of IoT service layers can also be virtualized. Several global standards (such as oneM2M) and personal (such as IBM Watson) IoT service sub-box platforms have integrated cloud and IoT to provide scalable IoT services using a slice sub-box. Data centers can perform complex computations and data analysis and are therefore responsible for reusing latency-tolerant services containing large amounts of storage and computing at the head end to improve the processing of edge computing tasks [14]. In particular, the IoT bias sniffs large amounts of data and transfers their services only to edge servers instead of unpacking them directly into balls to reduce the required signaling and corresponding energy consumption in the decision tree [20].

The concept of Multi-Access Edge Computing (MEC), as defined by the European Telecommunications Standards Institute (ETSI), is gaining traction with practical implementations. Given that network slicing and virtualization are fundamental to MEC, this discussion also incorporates the latest advancements in 3GPP technologies. These include mechanisms for slicing IoT service resources, which can be deployed on peripheral boards [21]. A game-theoretic approach called the Edge Resource Allocation (ERA) Game is used to address the challenge of pricing edge server resources owned by multiple stakeholders. This method delivers a solution that satisfies the conditions of a pure Nash equilibrium (PNE) for the ERA problem. By leveraging the ERA Game framework, the ERA algorithm is designed, allowing the system to converge at PNE. Once convergence nears, edge servers are divided into distinct groups, prompting the activation of the ERA algorithm. The algorithm operates concurrently across all edge servers within each cluster. It has been demonstrated that the ERA framework is a viable model, ensuring at least one PNE based on the ERA algorithm [22].

Furthermore, vulnerabilities within network processes are identified and addressed, emphasizing the importance of lifecycle management for resolving such issues. This is crucial for safeguarding digital twins and developing robust network distribution strategies. The study also highlights protective measures to enhance the security of Industrial IoT systems. A key innovation lies in applying game theory to analyze network security risks, offering fresh insights into effectively understanding and mitigating information security vulnerabilities in digital twin networks [23]. Although mobile edge computing can improve the efficiency of Mobile device (MD) applications, the simultaneous transmission of MDs can degrade the

TABLE 1: Comparison of recent game-theoretic approaches for resource allocation in MEC-enabled IoT systems.

Ref	Approach/Method	Key Idea	Strengths	Limitations
[2]	Game-Theoretical Task Allocation	Reward-driven for cognitive IoT	Optimized for user-specific needs	Might be complex for real-time applications
[3]	Game-Theoretical User Allocation	Edge computing environment	Decentralized control, user participation	Scalability under large loads?
[6]	MOACO + RL	Multi-objective optimization with RL	Good performance in IIoT	Training cost of RL
[9]	DRL-Based Resource Management	For Industrial IoT	Self-adaptive, high performance	May suffer from convergence delay
[10]	Caching & Multicast in 5G	Optimizes BS behavior	Better network efficiency	Limited real-time adaptability
[14]	Edge Intelligence & Energy Efficiency	Combines offloading + energy reduction	Scalable, effective for mobile devices	Needs careful balancing
[22]	ERA Game Model	Nash Equilibrium-based pricing	Fast convergence, fairness	Grouping overhead possible
[25], [26]	DDRM Algorithm with DDPG	Solves high-dimensional MDP	Reduced task arrival delays	High training complexity
[30]	DRL-Based MEC Task Scheduling	Optimizes task delay	Dynamic & adaptive	Initial model training required

channel quality. Although the clustering technique is used for wireless data transmission, previous computational decoding studies rarely used the concept of clustering to improve the efficiency of game-theoretic decoding [24].

A brand-new videotape analytics frame for blockchain-enabled Internet of autonomous vehicles (IoAV) with MEC script in which a videotape offloading and resource allocation problem is formulated to reduce the system's idleness and maximize the blockchain system's sale output. The possibility of an incorrect successive interference cancellation (SIC) has been investigated in non-orthogonal multiple access (NOMA) based IoT systems. An energized effectiveness optimization problem has been formulated to pair and allocate the radio resource.

An algorithm dynamic resource management (DDRM) was developed to explain the model for sequential stochastic decision problems (MDP), using Deep Deterministic Policy Gradient (DDPG) to handle, high dimensionality of state and action space. Experimental results showed that DDRM could effectively decrease task arrival rate compared to uniform resource management and random resource management algorithms [25], [26]. The socialization of resource sharing, value creation, user participation, supply-personalization, on-demand use, and demand matching in manufacturing run much more clearly and quickly. Wider applications of these Application management systems are hampered by a lack of open architecture, common specifications and standards, intelligent perception, and internet connectivity for the underlying physical manufacturing resources [27]. A novel process model might use fog computing. It brings cloud services and computing to the end of the network.

The IoT landscape consists of connections between different network anomalies. Virtual machine (VM) origin is far from an isolation scheme, as moving a virtual machine in real-time allows you to move an entire running task to another virtual machine. Based on this approach, Clone Pall and Cloudlet proposed computational offloading to Pall by running mobile polar tasks on remote virtual machines without programming [28], [29]. MEC collaboration in computing and communications is proposed by [30] which uses deep reinforcement learning-based dynamic resources management algorithm to lessen the long-term average delay of tasks to improve the performance of IoT.

The content caching mechanism is used to improve data delivery and its efficiency. A geographical cluster model is design for the retrival of content and algorithms are used to fine delays and transmission cost [31]. Caching methods are cetegroized based on their location, granularity, and coordination mechanisms, highlighting their effects on reducing latency and offloading core networks [32]. Artificial intelligence is impacting every domain in computing, cashing algorithms and techniques are also use machine learning and deep learning [33]. The [34] use deep learning base algorithms to optimizing resource allocation in vehicular networks. The proposed deep reinforcement learning model enables vehicles to act as intelligent agents that dynamically allocate communication resources based on environmental feedback and network conditions.

III. SYSTEM MODEL AND DESCRIPTION

The system architecture in Figure 2 of this paper has four layers: the cloud, the MEC layer, the user, and the IoT device
layer. The IoT device layer includes various gadgets like mobile phones, and smart IoT base environments which contain sensors and actuators that scan the environment and collect raw data. The user layer allows each user to control and process the IoT device's data. The MEC layer receives all the raw data and performs data pre-processing and analysis. The output data is reversed back to the user or to the cloud for more analysis and future use. The user should have an interactive real-time

TABLE 2: Description of symbols used in the equation.

Notation	Description	Notation	Description
BSs	Base station	b_k	Segments width
K	Number of Base stations	λ_k	Index
ECN	Edge Computing Nodes	μ_w	Average Service Rate
d_k	Service delay threshold time	D_r^w	Remaining Execution Time
τ_k^{th}	Quality of service requirements	D_t^w	Delay Time of edge
t_k^{net}	Represents network delay	η_w	Number of Segments
t_k^{comp}	Initial component delay time	λ	Average Arrival Rate

application that shows the data analysis results immediately. The MEC layer consists of a group of edge computing nodes (ECN), each with several low-power computing resources that can store computing devices.

Each BS is connected to an MEC server which acts as a small data center. The BS and the MEC server are in the same network and can cache and store content locally. This reduces the service latency and network congestion as the content is closer to the end users. Video streaming is one of the main applications that take advantage of this technology.

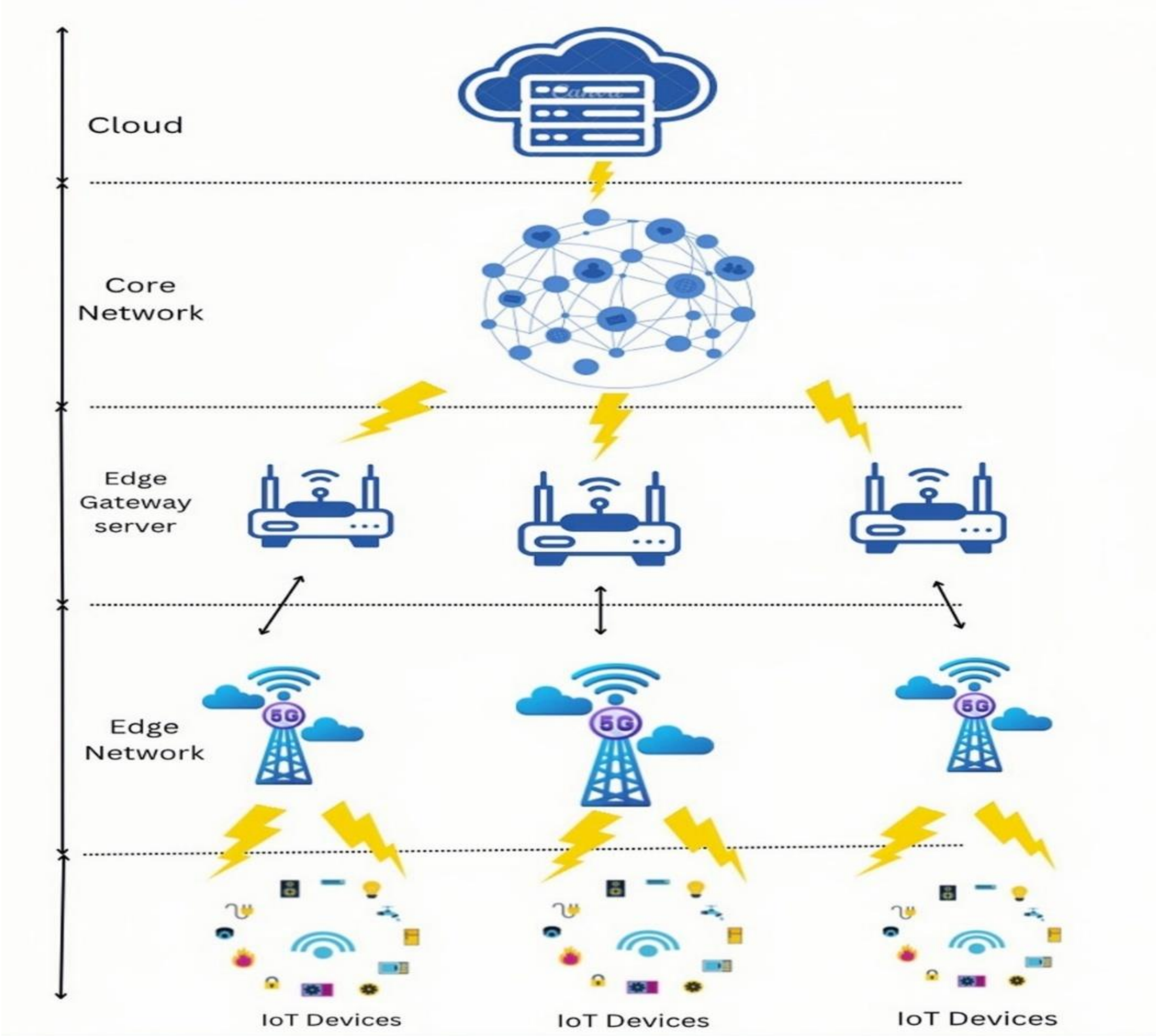

FIGURE 2: System model with four layers IoT Networks

When an end-user requests for a video stream with a given bit rate in which the MEC server works, it responds to the request first then the requested video stream is saved in the MEC server storage. Otherwise, it is requested to the cloud, which will take time increase the latency in response time, and use network supply to provide the desired video. Additionally, if the video is not cached at the requested bitrate and the video is cached at a higher bitrate, the request can also be satisfied by transcoding on the MEC server.

IV. PROBLEM FORMULATION

The problem formulation outlines a resource allocation strategy in IoT networks supported by MEC. To model computing capacity, the system uses Computing Resource Blocks (CRBs), while considering base station competition and latency constraints. Key parameters, including service delay thresholds, network delay, and the computing power of Edge Computing Nodes (ECNs), are clearly defined and incorporated into the model. Transmission latency is derived using Shannon's equation, accounting for time-varying channel conditions, and request patterns follow a Poisson distribution to reflect realistic traffic behavior. The formulation adopts queuing theory to estimate execution delays and system load, introducing metrics such as remaining service time and task queue length. The optimization challenge is initially non-convex but is shown to be convex under certain parameter conditions, which are explicitly stated. A game-theoretic model specifically, a Stackelberg game is employed to model the interaction between the MEC server and base stations. The proposed solution uses the Newton–Raphson method in an iterative algorithm, with each step of the algorithm, including initialization, sorting, and convergence criteria, thoroughly described to ensure reproducibility. The rationale behind selecting this model lies in its ability to balance fairness, utility maximization, and computational feasibility in a distributed network environment.

The MEC server can distribute Computing resources in units called CRBs. Each CBR can offer computing services at a rate of µ. Suppose that the storage capacity of MEC server is set as Q^s and CRB named as Q^c. Moreover, for improving the quality of service the range of M is M ∈ {1, 2, ..., M}. Which competes for the partial resources of BSs and MEC server of its users. User will communicate with their respective BSs through devices. If we have M number of users then λ_m is for m[th] BS users request. There are two categories for requesting the server one is for video service and other is for data service. We know that, 0 ≤ η ≤ 1, this is the presentation of proportional relation of video and service request. Therefore, the arrival rate of the entire video service request to the BS can be expressed as follows $\eta\lambda_m$.

Suppose that there are entirely K BSs identified as d_k and k ∈ {1, 2, . . ., K} and W, ECNs assigned by $f_w$here w ∈ {1, 2, . . ., W}. Other BSs have dissimilar calculation criteria that can be calculated using service lag. For example, particular BSs may select the lowest service delay at the cost of higher costs, while others may require the lowest cost at the cost of longer computations. The service delay threshold time d_k, required to meet the quality-of-service requirement is denoted by τ_k^{th}. In other words, the total delay in serving in the segment d_k, given as t_k, must satisfy the requirement $t_k \le \tau_k^{th}$. Now, the total delay for serving of section from BS to d_k, consists of both components,

$$\boldsymbol{t_k = t_k^{net} + t_k^{comp}} \quad \textbf{(1)}$$

Here the initial component t_k^{net} represents the network delay t_k^{comp}, which indicates the aggregate adjournment of together the delaying time and the service time.

The wireless channel involves the mobile edge caching and BS as a credible time-varying network, as a Finite Markov Channel (FSMC). The arriving SNR is shown γ_k^w, whose transition follows a Markov process. Therefore, the network latency t_k^{net} can expressed according to the Shannon equation, as, $^{o_k}/_{b_k}\ \log(1+\gamma_k^w)$ where $o_k$data represents the dimension for the segment, and b_k is the width.

The data segments of BSS d_k follow a Poisson division with an index λ_k, k = {1, 2, . , . ., K}. For edge computation, each ECN is considered to have different computing power and ECN f_w is assumed to be able to run computation service with an average service rate of μ_w. It is assumed that one data segment from BSS d_k is configured to serve by ECN f_w presenting smart contract. The calculation time of arc representation segments can be divided into two parts: The delay timef_w as part of the smart contract. The calculation time of arc representation segments can be divided into two parts: The delay time D_q^w and D_t^w, respectively. Therefore, the entire computation delay time of the edget_k^{comp}can be expressed as

$$\boldsymbol{t_k^{comp} = D_q^w + D_t^w} \quad \textbf{(1.5)}$$

The average computation time of the ECN f_w server data segment for the CPU can be calculated from the average service frequency, which is 1/µw. Then, equation (2) can be updated as follows equation (3) can be written as

$$\boldsymbol{t_k^{comp} = D_q^w + \frac{1}{\mu_w}} \quad \textbf{(2)}$$

$$\boldsymbol{t_k^{comp} = D_r^w + \frac{n_w}{\mu_w}} \quad \textbf{(3)}$$

where D_r^w is the remaining execution time of the segment on the server n_w show how many segments in the queue moving to the next segment d_k from the BSs is on time. For doing this job first-come, first-served technique is used, means each ECN is counted one shared in one time in the beginning of the queue. For making ECN load free we send the data to the cloud or base station when computing is complete for one of it. For convenience, the average time for processing is approx. D_r^w, which is shown below:

$$\boldsymbol{D_r^w = \frac{1}{2}\lambda\frac{1}{\mu_{w2}}} \quad \textbf{(4)}$$

The average arrival rate is the shown as λ and f_w. So, the entire delay time of edge calculation can be done on it as

$$\boldsymbol{t_k^{comp} = \frac{1}{2}\lambda\frac{1}{\mu_{w2}} + \frac{n_w}{\mu_w}} \quad \textbf{(5)}$$

Currently, the expression of the total serving delay f_w for one BSS segment d_k by ECN can be expressed as

$$\boldsymbol{t_k} = \frac{o_k}{b_{k\log(1+\gamma_k^w)}}\frac{1}{2}\boldsymbol{\lambda}\frac{1}{\mu_{w2}} + \frac{n_w}{\mu_w} \le \boldsymbol{\tau_k^{th}} \qquad \textbf{(6)}$$

It shows that the time spent on the validation activity is excluded from the absolute value of total time. After completing the calculation, the calculation is completed, the calculation result should be sent back to ECN f_w or stored in the cloud as soon as the calculation is finished. If the calculation results cannot be verified later ECN f_w tokens will not be received. But a certain percentage of the deposit will be deducted and BSS will be returned. However, a percentage of the deposit will be deducted, and BSs d_k

TABLE 3: Description of symbols used in the theorem.

Notation	Description	Notation	Description
p^c	Service Price	α_m^c	Weight factors of computing resource
q_m^{c*}	Amount of CRB	β_m^c	Utility function of BSs
CRB	Computer Resource Block	α_m^s	Weight factors of caching resource
q_m^s	Size Storage Capacity	β_m^c	Utility function of BSs
p^c	Caching price	o_k	Data segments dimensions
U_m^c	Quasi-Concave Function	$B_m(p^c)$	Best Computation Price

According to the modeled architecture of the network consider which games to allocate computing resources to. Consider the optimization problem q_m^{c*} as follows.

Theorem: if an MEC server advertises its service price p^c, the computer resource allocation model is likely to result optimally in the number of her CRBs with her B_m^c, denoted by q_m^{c*} there is,

$$\boldsymbol{q_m^{c*}} = \left[(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c\mu}{\beta_m^c p^c}} - \mathbf{1}\right)\right] \qquad \textbf{(7)}$$

with an $[\,] \triangleq \mathrm{maximum}(\,\cdot, 0)$.

Proof: From the utilization of function B_m^c, the first increase U_m^c with respect to q_m^c can be,

$$\frac{\partial U_m^c}{\partial q_m^c} = \frac{\alpha_m^c\mu}{[1+(q_m^c/(t_{th}-\theta d_m))]} - \boldsymbol{\beta_m^c\rho^c} \qquad \textbf{(8)}$$

also, the alternate outgrowth in it q_m^c is

$$\frac{\partial U_m^c}{\partial q_m^c} = -\frac{2\,\alpha_m^c\mu}{[1+(q_m^c/(t_{th}-\theta d_m))]^3\,(t_{th}-\theta d_m)} \qquad \textbf{(9)}$$

concerning as $\partial^2 U_m^c / \partial q_m^{c^2} < 0, \forall m\epsilon M$. therefore, U_m^c is a unique function concerning q_m^c.

its highest, i.e.,

$$\boldsymbol{q_m^{c*}} = \left[(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c\mu}{\beta_m^c p^c}} - \mathbf{1}\right)\right] \qquad \textbf{(10)}$$

thus, the responses of the BSs, we adjust that the problem for the MEC server is still extreme, thus, for the m^th BS, the index function to show

$$\sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c\mu}{\beta_m^c p^c}} - \mathbf{1}\right) \le \boldsymbol{Q^c}$$

(11)

$$\boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\,[\sqrt{\frac{\alpha_m^c}{\beta_m^c p^c}} - \mathbf{1}$$

m^th value of participates in the game.

$$\boldsymbol{y_m} = \begin{cases} \mathbf{1}, & \boldsymbol{p^c} < \frac{\alpha_m^c\mu}{\beta_m^c} \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \qquad \textbf{(12)}$$

Else (6) With the below index function for the Base station, optimization equation (5) is reformulated as

$$\max \boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M}\boldsymbol{y_m}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c\mu}{\beta_m^c p^c}} - \mathbf{1}\right)$$

(13)

$$\max \boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M}\boldsymbol{y_m}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\left(\sqrt{\frac{\alpha_m^c\mu}{\beta_m^c p^c}} - \mathbf{1}\right) \le \boldsymbol{Q^c} \qquad \boldsymbol{y_m} \in \{\mathbf{0}, \mathbf{1}\}$$

It's egregious that the below problem is because of the index, so it's not a convex problem. Nevertheless, it is not hard to prove that problem (7) is convex for a given index vector y. Equation (7) therefore assumes that Q^c is sufficient for entire BSs to participate to the game. As an outcome, all BS pointers are equivalent to 1.

$$\boldsymbol{p^c} < \left(\frac{\alpha_m^c\mu}{\beta_m^c p^c}\right), \forall \boldsymbol{m} \in \boldsymbol{M}, \qquad \textbf{(13.1)}$$

In this statement, the problem is curved and optimal. For solving this problem given below hypothesis could be follow. The best result of equation (7) pointers

(i.e. $y_m = 1, \forall m\epsilon M$) is given by

$$\boldsymbol{p^{c*}} = \mathbf{max}\{\boldsymbol{B_m}(\boldsymbol{p^c}), \boldsymbol{p^{c,LB}}\} \qquad \textbf{(14)}$$

where $B_M(p^c)$ satisfied

$$\boldsymbol{B_M}(\boldsymbol{p^c}) = \mathbf{arg}\,\underset{\boldsymbol{p^c}}{\boldsymbol{max}}\,\boldsymbol{U_{MEC}^c}, \quad \forall \boldsymbol{m} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{M} \qquad \textbf{(15)}$$

$$\boldsymbol{p^{c,LB}} = \left[\frac{\sum_{m=1}^{M}(t_{th}-\theta d_m)\sqrt{\frac{\alpha_m^c\mu}{\beta_m^c}}}{\sum_{m=1}^{M}(t_{th}-\theta d_m)+Q^c}\right] \qquad \textbf{(16)}$$

Proof

In expressed problem (7), we take the first outgrowth of

$$\boldsymbol{U_{MEC}^c} = \sum_{m=1}^{M}(\boldsymbol{p^c} - \boldsymbol{e^c})(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\,[\sqrt{\frac{\alpha_m^c}{\beta_m^c p^c}} - \mathbf{1} \qquad \textbf{(17)}$$

$$\frac{\partial U_{MEC}^c}{\partial p^c} = \sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\,(\tfrac{1}{2}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(1/2)}}}) + \sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m})\,\boldsymbol{e^c}(\tfrac{1}{2}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(3/2)}}}) - \sum_{m=1}^{M}(\boldsymbol{t_{th}} - \boldsymbol{\theta d_m}) \qquad \textbf{(18)}$$

also, the alternate outgrowth with respect to q_m^c is

$$\frac{\partial^2 U_{MEC}^c}{\partial p^{c2}} = -\sum_{m=1}^{M}(t_{th}-\theta d_m)\left(\frac{1}{4}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(3/2)}}}\right) - \sum_{m=1}^{M}(t_{th}-\theta d_m)\, e^c\left(\frac{3}{4}\sqrt{\frac{\alpha_m^c}{\beta_m^c p^{c-(3/2)}}}\right) \quad (19)$$

It's obviously seen that $\left(\frac{\partial^2 U_{MEC}^c}{\partial p^{c2}}\right) < 0, \forall p^c > 0$. Thus, $\frac{\partial U_{MEC}^c}{\partial p^c}$, this is a monotonically reducing function in the interval 0, ∞. Also, as the equation shows,

$$\lim_{p^c\to\infty}\frac{\partial U_{MEC}^c}{\partial p^c} = -\sum_{m=1}^{M}(t_{th}-\theta d_m) < 0 \quad \text{and} \quad \lim_{p^c\to 0}\frac{\partial U_{MEC}^c}{\partial p^c} = +\infty > 0 \quad (20)$$

is always decided. therefore, U_{MEC}^c has a definite maximum value. is using $B_M(p^c)$ to indicate the satisfied price (13). Based on the following analysis, an algorithm named Newton-Raphson is proposed by our model which obtain the upcoming value $B_M(p^c)$.

The total number of CRBs assigned to each base station must be in the capacity of the MEC Server's limited processing resources, the computing cost must satisfy inequality (8) price calculation.

$$p^{LB} = \left[\frac{\sum_{m=1}^{M}(t_{th}-\theta d_m)\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c}}}{\sum_{m=1}^{M}(t_{th}-\theta d_m)+Q^c}\right]^2 \quad (21)$$

Therefore, the optimal hidden cost determined by the MEC Server (9) can be attained. Judging from the given results, our model is able to work in general situations (7). Assuming all BSs are ordered, the algorithm also gives the optimal result for the problem.

Proposed Algorithm

Algorithm: An iterative algorithm based on the Newton–Raphson method $B_M(p^c)$

Step 1: Set K=M and $y_m = 1, \forall m \in M$

Step 2: Sort the K BSs according to $\frac{\alpha_m^c \mu}{\beta_m^c}$

i.e., $\left(\frac{\alpha_1^c \mu}{\beta_1^c} > \cdots > \frac{\alpha_{M-1}^c \mu}{\beta_{M-1}^c} > \frac{\alpha_M^c \mu}{\beta_M^c}\right)$

Step 3: Compute $B_M(p^c)$ based on $p^{c,k+1} = p^{c,k} - \frac{\partial U_{MEC}^c}{\partial p^c} \Big/ \frac{\partial^2 U_{MEC}^c}{\partial p^{c2}}$

$B_M(p^c) = p^{c,k}$

$$p^{LB} = \left[\frac{\sum_{m=1}^{M}(t_{th}-\theta d_m)\sqrt{\frac{\alpha_m^c \mu}{\beta_m^c}}}{\sum_{m=1}^{M}(t_{th}-\theta d_m)+Q^c}\right]^2$$

Step 4: $p^{c,best} = \max\{B_M(p^c), p^{c,LB}\}$

Step 5: Compare the $p^{c,best}$ with $\frac{\alpha_K^c \mu}{\beta_K^c}$:

If $p^{c,best} < \frac{\alpha_K^c \mu}{\beta_K^c}$ then

go to step 3.

Step 6: $p^{c,Optimal} = p^{c,best}$

The SE realized by the proposed system enables well-balanced utility usage among the mobile edge caching server and numerous IoT base stations.

V. RESULTS AND ANALYSIS

This section presents the results of our proposed model. We simulate our outcomes on MATLAB that give a rich and precise yields. While doing simulated the IoT-based scenario without explicit clarification, we can observe that maximizing the capacity of the MEC server storage for more benefits. As it grows, the utility increases, until reaching a certain point where further increases do not bring any extra advantage. The simulation was conducted in an environment consisting of 25 BSs and with a calculated computation capacity of 50 all having the same Zipf distribution characteristic (τ at 0.5). Every base station's request arrival rate is randomly set to average 10 ms 1 with video service requests comprising half of this rate. On average, each BS can cache 500 videos. Additionally, as per [29], Each CRB has a service rate of 0.1 minutes per second, but all 25 BSs have a delay tolerance of 60 ms.

The storage capacity has a major influence on the cache price. When it is limited, competition among BSs is so furious that the cost of cache is extremely high. However, as the storage capacity increases, the cache price gradually decreases until reaching an equilibrium point (i.e., when pLB ≤ e s / (1 − τ)). Upon attaining this point, the cache fee no longer relies on storage capacity and remains constant.

As indicated in Figure 3, given a storage size of 50, computation power has a positive effect on MEC server utility, raising initially and then stabilizing to a certain value. This occurs as more CRBs are available for allocation to the BSs. Furthermore, with an increased number of BSs and computation level remaining equal, the heightened rivalry between BSs causes converged value to grow higher.

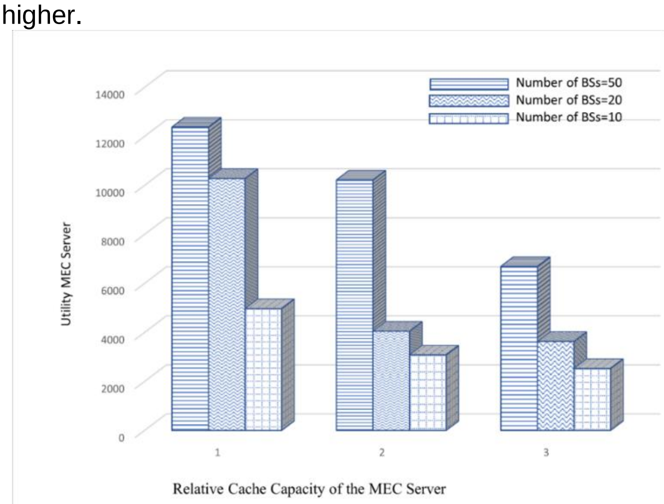

FIGURE 3: Influence of cache capacity on utility MEC server for different sets of BSs

This research evaluates the performance of the mobile edge caching server against its capacity by presenting simulation results. The transmission distance between it and its base stations is a randomly generated number between 0 and 10 km, while the rent follows a random uniform distribution between 1 and 25. Furthermore, the average weight factors for the server and CRBs are α / β = 50/0.2 and α s/ βs= 500/0.1, respectively. Figure 4 shows our findings.

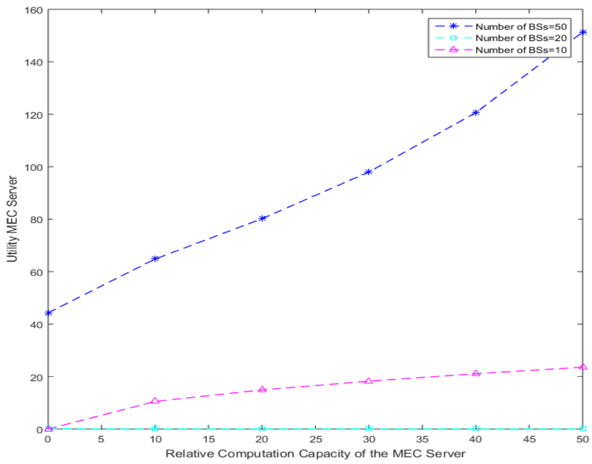

FIGURE 4: Variation in Utility MEC Server with computation capacity

The game-based scheme proposed by Stackelberg is compared with other two methods.
(i) The first one is the YM (Yield Management) Approach which sees the MEC server offering discounts on available resources increments of 30 regarding cache and computation capacity comes with a 5% discount on the current price.
(ii) Second is the greedy scheme, in which MEC acts like a monopolist and enhances overly expensive prices.

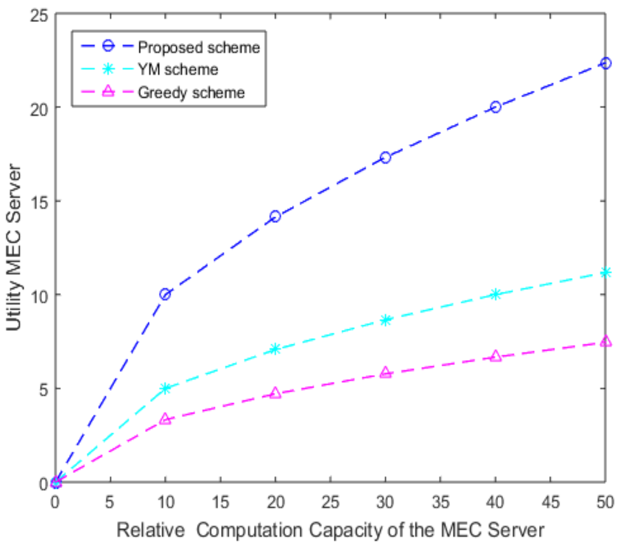

FIGURE 5: Relative computation capacity behaviour of MEC Server under diverse patterns

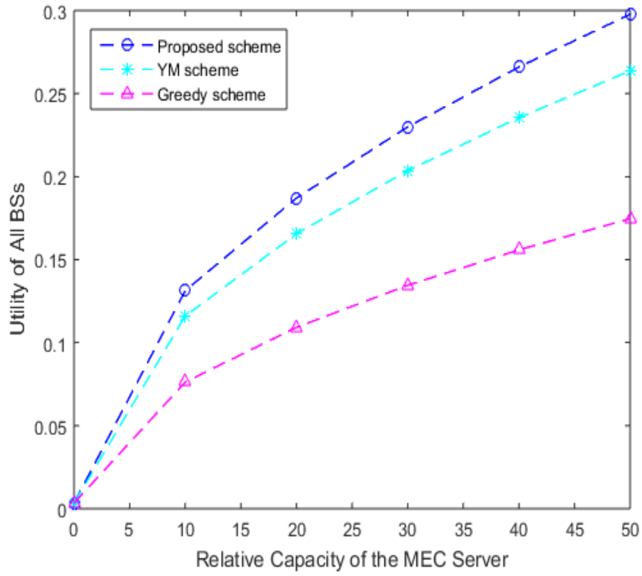

FIGURE 6: BSs efficacy with the capability of MEC server

According to Figures 5 and 6, the utility of MEC servers and BSs are determined by three pricing models concerning resource capacity, respectively. A greedy system allows MEC servers to catch up with Stackelberg's game-based system, as MEC servers always offer the best prices. Figure 6 proves that the SE completed by the suggested system can well balance utility among the MEC servers and numerous IoT base stations.

Figure 7 shows the utility estimation of the MEC server when the service speed μ alters from 0.1 to 1 with a step size of 0.1. Utility increases by μ. Here's why a larger value of μ allows the MEC server to handle more requests directly instead of forwarding them to withdrawn servers within an acceptable service delay for IoT. Therefore, MEC servers can generate more revenue from base station requests.

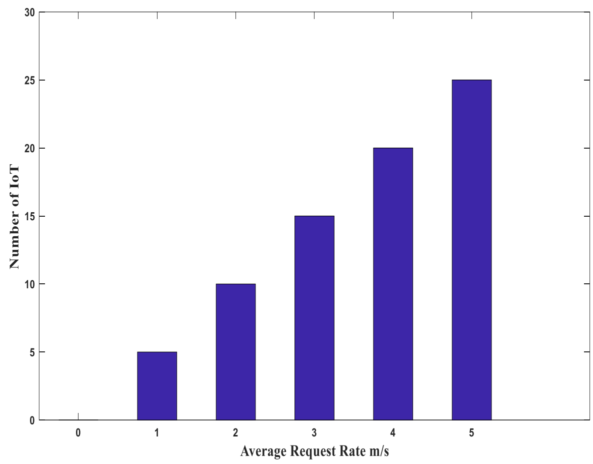

FIGURE 7: Number of IoTs versus the request rate

This research evaluates the performance of the MEC server against its capacity by presenting results. The transmission distance between it and its base stations is a randomly generated number between 0 and 10 km, while the rent follows a random uniform distribution between 1 and 50. Furthermore, the average weight factors for the server and CRBs are α / β = 50/0.2 and αs/ βs= 500/0.1, respectively.

VI. CONCLUSION

Our research proposes a framework named MEC for resource allocation servers and base station connections to ensure efficient resource planning of IoT cellular networks. The proposed algorithm significantly increases the system efficiency and reduces the response time. The simulation results show the effectiveness of the proposed system. In addition, the originality and equilibrium connection of the Stackelberg game and the reverse induction system are proposed as a solution to the resource allocation problem. For future work, we encourage you to consider efficient computation offload strategies for cross-IoT collaboration. Our future work integrates with neural networks for better analysis of the system and adds predictive capabilities. Adding experimental tests makes it easier for users. Multiple optimization objects are another promising direction for improving overall network performance and are also considered as a new research direction for future research.

REFERENCES

[1] Annual, C., & Report, I. (2018). *White paper Cisco public*.

[2] Rahman, T. F., Pilloni, V., & Atzori, L. (2019). Application Task Allocation in Cognitive IoT: A Reward-Driven Game Theoretical Approach. *IEEE Transactions on Wireless Communications*, *18*(12), 5571–5583. https://doi.org/10.1109/TWC.2019.2937523

[3] He, Q., Cui, G., Zhang, X., Chen, F., Deng, S., Jin, H., Li, Y., & Yang, Y. (2020). A game-theoretical approach for user allocation in edge computing environment. *IEEE Transactions on Parallel and Distributed Systems*, *31*(3), 515–529. https://doi.org/10.1109/TPDS.2019.2938944

[4] Kim, S., Cai, H., Hua, C., Gu, P., Xu, W., & Park, J. (2020). Collaborative Anomaly Detection for Internet of Things based on Federated Learning. *2020 IEEE/CIC International Conference on Communications in China, ICCC 2020*, 623–628. https://doi.org/10.1109/ICCC49849.2020.9238913

[5] Tang, Q., Xie, R., Huang, T., & Liu, Y. (2019). Jointly caching and computation resource allocation for mobile edge networks. *IET Networks*, *8*(5), 329–338. https://doi.org/10.1049/iet-net.2018.5111

[6] Vimal, S., Khari, M., Dey, N., Crespo, R. G., & Harold Robinson, Y. (2020). Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT. *Computer Communications*, *151*, 355–364. https://doi.org/10.1016/j.comcom.2020.01.018

[7] Zhang, J., Hu, X., Ning, Z., Ngai, E. C. H., Zhou, L., Wei, J., Cheng, J., Hu, B., & Leung, V. C. M. (2019). Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching. *IEEE Internet of Things Journal*, *6*(3), 4283–4294. https://doi.org/10.1109/JIOT.2018.2875917

[8] Ning, Z., Dong, P., Kong, X., & Xia, F. (2019). A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things. *IEEE Internet of Things Journal*, *6*(3), 4804–4814. https://doi.org/10.1109/JIOT.2018.2868616

[9] Chen, Y., Liu, Z., Zhang, Y., Wu, Y., Chen, X., & Zhao, L. (2021). 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. https://doi.org/10.1109/TII.2020.3028963

[10] Poularakis, K., Iosifidis, G., Sourlas, V., & Tassiulas, L. (2016). Exploiting Caching and Multicast for 5G Wireless Networks. *IEEE Transactions on Wireless Communications*, *15*(4), 2995–3007. https://doi.org/10.1109/TWC.2016.2514418

[11] Bandopadhyay, A., Mishra, V., Swain, S., Chatterjee, K., Dey, S., Mallik, S., ... & Soufiene, B. O. (2024). EdgeMatch: A Smart Approach for Scheduling IoT-Edge Tasks With Multiple Criteria Using Game Theory. IEEE Access.

[12] Hu, P., Ning, H., Qiu, T., Zhang, Y., & Luo, X. (2017). Fog computing based face identification and resolution scheme in internet of things. *IEEE Transactions on Industrial Informatics*, *13*(4), 1910–1920. https://doi.org/10.1109/TII.2016.2607178

[13] Antonius, F. (2024). Efficient resource allocation through CNN-game theory based network slicing recognition for next-generation networks. Journal of Engineering Research.

[14] Dai, Y., Zhang, K., Maharjan, S., & Zhang, Y. (2020). Edge Intelligence for Energy-Efficient Computation Offloading and Resource Allocation in 5G beyond. *IEEE Transactions on Vehicular Technology*, *69*(10), 12175–12186. https://doi.org/10.1109/TVT.2020.3013990

[15] Cuervo, E., Wolman, A., Cox, L. P., Lebeck, K., Razeen, A., Saroiu, S., & Musuvathi, M. (2015). Kahawai: High-quality mobile gaming using GPU offload. *MobiSys 2015 - Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 121–135. https://doi.org/10.1145/2742647.2742657

[16] Chen, W., Wang, D., & Li, K. (2019). Multi-User Multi-Task Computation Offloading in Green Mobile Edge Cloud Computing. *IEEE Transactions on Services Computing*, *12*(5), 726–738. https://doi.org/10.1109/TSC.2018.2826544

[17] Chu, Z., Xiao, P., Shojafar, M., Mi, D., Mao, J., & Hao, W. (2021). Intelligent Reflecting Surface Assisted Mobile Edge Computing for Internet of Things. *IEEE Wireless Communications Letters*, *10*(3), 619–623. https://doi.org/10.1109/LWC.2020.3040607

[18] Feng, H., Chen, D., Lv, H., & Lv, Z. (2023). Game theory in network security for digital twins in industry. *Digital Communications and Networks*. https://doi.org/10.1016/j.dcan.2023.01.004

[19] Huang, Y. Y., & Wang, P. C. (2023). Computation Offloading and User-Clustering Game in Multi-Channel Cellular Networks for Mobile Edge Computing. *Sensors*, *23*(3). https://doi.org/10.3390/s23031155

[20] Ibrahim, A. M., Chen, Z., Eljailany, H. A., Yu, G., Ipaye, A. A., Abouda, K. A., & Idress, W. M. (2024). Advancing 6G IoT networks: Willow Catkin packet transmission scheduling with AI and Bayesian game-theoretic approach-based resource allocation. Internet of Things, 101119.

[21] Kim, Y., Song, C., Han, H., Jung, H., & Kang, S. (2020). Collaborative Task Scheduling for IoT-Assisted Edge Computing. *IEEE Access*, *8*, 216593–216606. https://doi.org/10.1109/ACCESS.2020.3041872

[22] Kumar, S., Gupta, R., Lakshmanan, K., & Maurya, V. (2022). A Game-Theoretic Approach for Increasing Resource Utilization in Edge Computing Enabled Internet of Things. *IEEE Access*, *10*, 57974–57989. https://doi.org/10.1109/ACCESS.2022.3175850

[23] Li, X., Liu, Y., Ji, H., Zhang, H., & Leung, V. C. M. (2019). Optimizing resources allocation for fog computing-based internet of things networks. *IEEE Access*, *7*, 64907–64922. https://doi.org/10.1109/ACCESS.2019.2917557

[24] Premsankar, G., Di Francesco, M., & Taleb, T. (2018a). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

[25] Wang, T., Qiu, L., Sangaiah, A. K., Liu, A., Bhuiyan, M. Z. A., & Ma, Y. (2020). Edge-Computing-Based Trustworthy Data Collection Model in the Internet of Things. *IEEE Internet of*

Things Journal, *7*(5), 4218–4227. https://doi.org/10.1109/JIOT.2020.2966870

[26] Xue, H., Huang, B., Qin, M., Zhou, H., & Yang, H. (2020). Edge Computing for Internet of Things: A Survey. *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 755–760. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00130

[27] Premsankar, G., Di Francesco, M., & Taleb, T. (2018b). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

[28] Zhang, G., Zhang, S., Zhang, W., Shen, Z., & Wang, L. (2021). Joint Service Caching, Computation Offloading and Resource Allocation in Mobile Edge Computing Systems. *IEEE Transactions on Wireless Communications*, *20*(8), 5288–5300. https://doi.org/10.1109/TWC.2021.3066650

[29] Zhang, H., Xiao, Y., Bu, S., Niyato, D., Yu, F. R., & Han, Z. (2017). Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching. *IEEE Internet of Things Journal*, *4*(5), 1204–1215. https://doi.org/10.1109/JIOT.2017.2688925

[30] Zhang, K., Leng, S., He, Y., Maharjan, S., & Zhang, Y. (2018). Mobile Edge Computing and Networking for Green and Low-Latency Internet of Things. *IEEE Communications Magazine*, *56*(5), 39–45. https://doi.org/10.1109/MCOM.2018.1700882

[31] Kabir, A. (2018). Cooperative Content Caching and Distribution in Dense Networks. KSII Transactions on Internet & Information Systems, 12(11).

[32] Kabir, A., Rehman, G., Gilani, S. M., Kitindi, E. J., Ul Abidin Jaffri, Z., & Abbasi, K. M. (2020). The role of caching in next generation cellular networks: A survey and research outlook. Transactions on Emerging Telecommunications Technologies, 31(2), e3702.[1][w]

[33] Kumar, Y., Marchena, J., Awlla, A. H., Li, J. J., & Abdalla, H. B. (2024). The AI-Powered Evolution of Big Data. Applied Sciences, 14(22), 10176. https://doi.org/10.3390/app142210176

[34] Ergün, Serap. "Resource allocation optimization for effective vehicle network communications using multi-agent deep reinforcement learning." *Journal of Dynamics and Games* 12.2 (2025): 134-156.