TASK OFFLOADING AND RESOURCE ALLOCATION IN MOBILE EDGE CLOUD COMPUTING: REVIEW

Dr. R.Kaniezhil,
Principal,
Navarasam Arts and Science College for Women,
Arachalur Erode Dt.

K.Sujitha
Research Scholar,
Navarasam Arts and Science College for Women,
Arachalur Erode Dt.

ABSTRACT

Task offloading(TO) is invaluable in facilitating interaction between local devices and cloud infrastructure in areas such as AI, machine learning, and big data analysis. Its relevance is underpinned by the need and the necessity for devices that, themselves, do not possess significant computing capabilities, including but not limited to mobile phones, devices of the Internet of Things, and edge devices, to access computational resources. These techniques shifts the computations to cloud or edge servers from the devices that may be resource limited such as the mobile phones, IoT devices in cloud computing. Especially in cases of mobile or IoT, computations offloading helps in saving lots of energy required for intensive computation at the device. The cloud services that enable a task to be done flexibly in a given time without overloading local applications. These approaches are to enhance the performance, minimize the delay and regulate power usage. Outsourcing to the cloud may enhance task execution because the cloud. In terms of benefit it provides better processing power and the ability to scale to the level of demand. This means that organizations can avoid expending large sums in the purchase of hardware and other support structures by hiring cloud resources on as-needed basis. However, these techniques have many open challenges: The proposed arguments explain that (1) outsourcing tasks on to a cloud primarily depends on network quality and high latency hampers real-time or time-constrained tasks, (2) the offloading of sensitive tasks increases questions over data privacy and security, and (3) resource management is a challenge, especially when operating over multiple cloud domains. Thus, the rectification of the mentioned challenges in offloading forms the goal of

this study in a bid to enhance the offloading performance and reduce on energy consumption and cost.

Keywords:

Cloud Computing, Edge Computing, Energy Efficiency, Resource Management, Task Offloading

I. INTRODUCTION

The fog computing architecture, which is primarily composed of the edge layer, fog layer, and cloud layer, has been a study focus due to the widespread usage of wireless communication technology and the ongoing advancements in IoT technology [1]. A number of variables, including time-varying radio channels, available processing resources, and the devices' geographic location, are expected to have an impact on the choice to offload tasks [2], which may result in higher energy consumption and longer reaction times. Making the right offloading choices is challenging since edge-cloud computing includes changing resources and restricted computing resources. In mobile edge cloud computing (MEC), a paradigm that connects services at the network's edge to bring the cloud closer to consumers, task offloading and resource allocation are fundamental requirements [3]. Their responsibilities in this setting include increasing productivity and throughput, proactively attending to user demands, and optimizing the resources available within the heterogeneous and decentralized system.

Table 1: key characteristics of task offloading techniques

Offloading Technique	Key Benefits	Challenges	Common Use Cases
Static Offloading	Simple, predictable performance	Lack of flexibility	Predictable environments (e.g., certain IoT systems)
Dynamic Offloading	Flexible, adapts to changing conditions	Complex to implement	Mobile apps, IoT systems
Full Offloading	Maximum resource conservation	Dependent on network quality	Heavy computation tasks (e.g., data analysis)

Offloading Technique	**Key Benefits**	**Challenges**	**Common Use Cases**
Partial Offloading	Balances local and cloud resources	Task partitioning complexity	AR, AI, and hybrid applications
Computation Partitioning	Fine-grained control	Dependency management	AR, machine learning, real-time processing
Opportunistic Offloading	Leverages favorable network conditions	Inconsistent performance	Smart cities, vehicular networks
Collaborative Offloading	Distributes load across devices	Coordination complexity	VANETs, P2P systems
Cloudlet-based Offloading	Low latency, reduced bandwidth usage	Limited resources	AR, real-time video processing
Code Offloading	Reduces data transfer	Complex implementation	Mobile apps, resource-intensive functions
Energy-aware Offloading	Optimizes battery life	Network dependency	Mobile cloud computing, IoT

Each technique is selected based on the specific needs of the application [4][5], network conditions[6], and the device's capabilities.

Importance of Task Offloading and Resource Allocation in Mobile Edge Cloud Computing Environment

Below are the key reasons why task offloading and resource allocation are crucial in MEC:1. Enhanced performance and lesser latency Proximity to Users: MEC [7] decreases latency since application computations are near the user, unlike traditional cloud computing, where data could travel a great distance to data centers. The relocation of tasks to the edge nodes allows for their performance in a shorter period ideal for real-time and low latency applications such as augmented reality (AR), virtual reality (VR), gaming, and autonomous vehicles.

Edge-Based Processing: Specifically, more computationally intensive tasks, for example video processing, real time data processing, AI inferencing, can be executed on powerful edge servers thereby improving application and system response times.

2. The seventh theme is energy efficiency and battery conservation It refers to the fact that here in most of the solutions, a) it is proposed that the battery power utilization should be aggressively minimized and b) there are descriptions of how energy efficiency should be integrated into the solutions in general.

Offloading Heavy Computation[8]: Mobile and IoT devices have constrained resources including computational capacity and energy reservoir. Applications can lower the amount of computational load placed on a device, and thus save power and utilize battery in more efficient manner if part of this computation is performed by edge servers, stored in devices themselves. That is particularly relevant tobattery-operated products like smart phones, drones, and wearables among other digital devices.

Energy-Aware Resource Allocation: resources are shifted dynamically at the edge, so the processing of the tasks occurs at the points which are most efficient in terms of the energy consumption both at the end device level and at the edge server level. This is good for sustaining high performance of the systems while in energy-restricted environments, high energy-consumption is discouraged.

3. Availability, Accessibility and Utilization of Resources Efficient Resource Utilization: Some of the resource allocation mechanisms are as follows: Resource allocation P2P ensures that the scarce resources available at the edge including CPU, memory and storage are optimally used. By distributing work loads, resources such that it is evenly spread among multiple edge servers, the congestion head is easily avoided and also there is no wastage of resources. Dynamic Scaling: There are strategies for responding to changes in demand that then apply the optimal amount of resources required at the edge level. This makes sure that the system works adequate for increased amounts of data or other loads without suffering in performance, as clouds do.

4. Support of Real-time and Low Latency цями Low-Latency Services: Self-driving cars[8], intelligent health care, industrial applications, smart city services need real-time data processing where delay is not tolerated. These applications are realized by task offloading which means that data can be processed at the nearby edge nodes instead of on the remote cloud servers in order to satisfy strict latency demands.Reduced Network Congestion: Offloading from the core network to edge processing alleviates congestion in the core by less data requisite to pass through the core network towards centralized DCs, as well as it entails less latency for exigent use cases.

5. Higher Order Quality of Experience (QoE) User-Centric Optimization: Based on the requirement of the users, the offloading of the tasks and the applications towards MEC environments[9] can be designed for individual wants and maximize the Quality of Experience (QoE) of every user. applicants enjoy improved response time, enhanced smoothness of the application under use and the quality of service delivered where demands for low latency and high throughput are involved.

Adaptive Resource Management: Given that all offloading decisions and resourceassignments are managed dynamically and according to real-time factors such as network conditions, the mobility of the user, and the requirements of the application, the QoE of the user remains high consistently.

6. Mobile Full and IoT EcosystemsLightweight Devices: MEC also allows advanced applications [5] and Big Data processing to be delivered and controlled by the mobile and IoT devices themselves with comparatively little need for local processing power. Such superiors hold multiple tasks to edge servers; these devices can perform advanced functions such as machine learning or real-time monitoring, which require simple computations, alongside augmented reality, using least power and costs of the hardware.

IoT Scalability: MEC facilitates this process by assisting on handling the huge data load that IoT billions of devices produce by offloading task to local edge servers. Such a distributed approach enables analyzation of data closer to its origination, and guarantees scalability in IoT landscapes.

7. Cost Efficiency Reduced Infrastructure Costs:

Decentralization using edge computing reduces the demand for high performing and costly hardware devices on mobile clients. Rather, various edge computing resources, which may be utilized by several users or applications, perform these tasks. This makes the costs for end-users and businesses less because it minimizes the requirement for localized supporting systems. Pay-As-You-Go Models: MEC environments[10] are normally priced akin to cloud services, meaning users only pay for the service used. This results in cost effectiveness because resources can be made more flexible and can be charged back upwards to the user level based on consumption level.

8. The following parameters mean support for AI and Data Intensive Applications Edge AI: Real-time inference based AI applications can massively benefit from edge based Task-Off-Loading. In some cases, such as smart city or industrial IoT, edge servers can take decisions based on the processed AI models and give immediate answer to queries without involving expensive cloud servers. Data-Driven Applications: The offloading of tasks that require the management of a large volume of data (video analysis, sensor data, image recognition, etc.) are carried out on the edge and frees up those applications from verburdening handheld devices or cloud networks.

9. Enable and Enhance Accessibility & Use Experience Seamless Task Migration: In mobile edge scenarios user can switch between the network locations (where it is applicable e.g. vehicular networks or mobile applications). The frameworks for offloading tasks and distributing resources that consider mobility of users avoid handover situations by managing their continuity in collaboration between different edge servers. Mobility-Aware Offloading: Mobile edge computing (MEC) offloading schemes[11] for handling user mobility are described as follows: It enables prediction of the mobility and reassigns the resources or tasks to the nearby edge servers to reduce the service interruption and at the same time to minimize the latency.

10. Privacy and Security Improvement Localized Data Processing: By loading certain tasks to the edge, nearer to the source of data, there is less emphasis on shifting large pieces of information over great distances to complex cloud servers. This makes the process more secure because data

does not have to travel long distances and they can be processed on safe edge nodes.Security-Aware Resource Allocation: MEC[12] environments can employ resource allocation policies to place sensitive tasks within secure and trusted contexts while achieving necessary and high performance. 11. 5G and Future Networks Making the Most of 5G: With 5G networks, users will have access to ultra-low latency and high bandwidth, which will allow them to offload tasks and allocate resources more efficiently. New applications like smart mobility, augmented and virtual reality, and telemedicine may be made possible with the aid of MEC, which moves certain duties to the edge nodes that are closer to 5G base stations. Network Slicing and Resource Isolation: In MEC, the resource allocation can also leverage on network slicing of 5G in that it provides various capabilities of the network but with different characteristics such as the latency and throughput of the network. It also lets resource provisioning be application-specific, enhancing the general quality of the service. Resource management and workload partitioning are two fundamental aspects of mobile edge cloud computing that augment application performance, reduction of power consumption, and emphasizing scalability. Real-time application, low latency applications, lightweight and mobile devices, in general, improving the quality of service to user and efficient utilization of resource, load balancing, intelligent task offloading are some of the benefits that are fulfilled by them. At a growing centre stage, these techniques would be relied on more in the formulation of the future generation applications among them: IoT, Artificial Intelligence, 5G, and more to come).

II. TASK OFFLOADING AND RESOURCE ALLOCATION IN EDGE COMPUTING

Ullah, I. et al. (2023) provide methods for controlling offloading and resources in order to reduce latency and fulfill the demands of computation and communication in edge-cloud computing. Another big challenge in edge-cloud computing systems is TO, a complex multi-objective optimization issue. To overcome this problem, we propose deep reinforcement learning. For this, we use the Double Deep Q-Network (DDQN) and model the problem as a Markov decision process. The proposed DDQN-edge-cloud (DDQNEC) approach adapts to the present system resource utilization, task limitations, and edge-cloud network circumstances to decide whether to offload an AI job to the cloud to reduce reaction time.

Also, the original problem may be divided into two pieces, according to T. X. Tran et al. (2019): the RA problem and the task offloading problem. The task offloading problem involves maximizing the optimal-value function that corresponds to the RA problem with the fixed TO option. The RA problem is solved using convex and quasi convex optimization, while the TO problem is solved using a heuristic in polynomial time, almost optimally.

Given the interconnected nature of offloading decisions with other activities, it is challenging to ascertain the exact optimal approach to task offloading and resource allocation. Assuming offloading settings are given, J. Yan et al. (2020) provide closed-form representations of the offloading transmit power and local CPU frequencies. Finally, the optimal solutions to the model are found using an efficient bi-section search approach. In addition, we show that a one-climb strategy is followed by the optimal offloading decisions. An easier method than the previous Gibbs Sampling methodology is provided for obtaining the optimum offloading selections. We next generalize our findings to a generic multi-user context by incorporating the outcomes of many final tasks performed at separate WDs into a single task executed at a single WD.

Quality of the channel, processing capability of the server, and load on the server were all factors in the user-server association technique employed by C.-F. Liu et al. (2019). A two-timeframe approach is detailed here, combining methods from matching theory and Lyapunov optimization; on the long timescale, it has a static user-server interaction; on the short timescale, it has a dynamic strategy for work offloading and resource allocation.

The JORA system, introduced by H. Jiang et al. (2023), ensures long-term MEC energy constraint while optimizing end-user quality of experience (QoE). Optimal solutions to the long-term QoE maximization issue may be found by the use of Lyupanov optimization, which is necessary for this goal. We can address this scenario's issue instantly since it's possible to create an energy shortage queue to stimulate more energy usage. This is the basis of the proposed online JORA approaches in both distributed and centralized ways. In addition to limiting MEC's overall energy consumption in the long run, the methods help achieve near-ideal performance.

With the help of separable SDR, M.-H. Chen et al. (2018) solved MUMTO without CAP and found out how to divide up communication resources and which binary offloading method to apply. To address this more complex problem, we provide an efficient MUMTO-C approach that integrates sequential tuning, the generalized alternating optimization process, and MUMTO SDR with CAP. In every case, a locally optimal solution is used to calculate the method.

In their 2018 study, Elgendy, Ibrahim A. et al. proposed a new method for balancing loads across ground-based MEC servers during handover and for UAVs flying over densely populated regions to do the same, all while keeping the GBSs' base locations consistent (2018). Additionally, it provides an integer description of the coupled problem of task offloading, load balancing, and resource allocation with the aim of reducing the system's cost. An efficient method of task offloading that makes use of deep reinforcement learning algorithms is also offered as a solution to this problem.

Resource allocation and offloading are examined in a densely deployed C-RAN that is based on MEC by Q. Zhang et al. (2020), with an eye towards energy efficiency. A chance-constrained MINLP modeling the task of selecting the offloading work, allocating computer resources, and assigning radio resources. Using the principles of Lyapunov optimization, the design problem is partitioned into four subproblems that may be solved to address the described difficulty. To tackle the lesser issues, we use convex decomposition techniques and matching games. Research conducted at the facility demonstrates the compromise between energy efficiency and the time it takes to get a service.

J. Almutairi et al. (2022) suggests a new approach to work offloading in multi-tier edge-cloud environments with multiple users, focusing on latency optimization. The proposed effort is an Integer Linear Programming (ILP) formulation with the overarching objective of reducing the total amount of time needed for UAV service.

T. Alfakih et al. (2020) suggests using an RL-SARSA algorithm—a reinforcement learning-based state action reward state action—to minimize system cost, which includes energy use and computational time delay—and to solve the edge server resource management problem. This method of extraction is known as offloading decision based SARSA (OD-SARSA).

III. TASK OFFLOADING AND RESOURCE ALLOCATION IN FOG ENVIRONMENT

The GOMOTO approach is proposed by Qian Ren et al. (2022) to address the multi-objective optimization problem, and they use network calculus theory to build an analytical performance model for task offloading in a fog computing environment. It distributes them to a large number of collaborating Fag Nodes (FNs) in order to carry out tasks.

The following tuple: latency, energy utilized, time deadline, and priority were incorporated in the MDP-based task offloading problem formulation by Jain, V. et al. (2023) as user QoS criterion. The goal of this paper is to provide three separate model-free off-policy Deep Reinforcement Learning (DRL) based algorithms for optimizing the reward to resource ratio.

To achieve this goal, H. Tran-Dang et al. (2021) provide FRATO, a comprehensive framework for IoT-fog-cloud systems. Service delivery delays are minimized by its excellent work offloading mechanism. Also, it employs a collaborative work offloading method that is directly linked to the data fragment idea, and it leverages the fog resource to determine the optimum offloading way dynamically. In addition, two distributed fog resource allocation algorithms, TPRA and MaxRU, are developed to efficiently implement offloading systems that have been tuned for use in situations where there is competition for resources.

Mobility task scheduling offloading is one component of the four-stage MISSION strategy developed by Matrouk, K.M. et al. in 2023. This method of handover takes previous data into account and deals with the dispersion of Internet of Things devices in order to decrease retransmissions. This is accomplished by 5G gateways using our RSS, Direction, and Distance parameter-based Mobility Aware Proximal Policy Optimization method (2). We employ a variety of criteria to sort the jobs into four groups, and then we use those groups as input for our work scheduling system. Di-Process Modular Neural Networks (Di-MNNs) have achieved two conceptual goals in the realm of neural networks: scheduling and classification. (3). energy-efficient work allocation based on the weight parameter of the job as determined by First Fitness Base

Animal Migration Optimization (FFAMO). A weighted work is then assigned to the best fog using the capacity-based Hungarian Assignment algorithm (CH2A), which takes 4. When the central fog node is overloaded with scheduling or job allocation, it exploits graph entropy to create a virtual fog.

As a three-objective optimization, task offloading aims to minimize total power consumption and task execution delay (Keshavarznejad, M. et al., 2021). Then, two meta-heuristic algorithms, namely the Bees algorithm and the non-dominated sorting genetic algorithm (NSGA-II), are used since the task is NP-hard.

S. D. Okegbile et al. (2022) propose a multi-user, multi-class, and multi-layer edge computing system to enable computation frameworks and task offloading. The three-tiered architecture they proposed would have devices transmit the tasks they finished in each time slot to a specific layer for processing. If the queue size is less than the threshold, the work is processed at this selected tier; otherwise, it is sent on to the next layer. Their evaluations are based on their classifications so that they may carry out responsibilities according to their degree of service.

For Internet of Things (IoT) sensor applications in a fog environment, Kishor, A. et al. (2022) introduced a meta-heuristic scheduler task offloading approach named Smart Ant Colony Optimization (SACO). These algorithms are compared to Bee life, modified particle swarm optimization, throttled scheduler, and round robin, and their outcomes are analyzed.

In their approach, F. Chiti et al. (2018) distributedly aims to optimize stability and matching quality by having each party make a proposal to the other candidates using the postponed acceptance algorithm. By calculating the worst-case total completion time, mean total completion time per work, and mean waiting time, computer simulations evaluate the proposed strategy's performance.Wu et al. (2020) describe the problem of offloading strategies that occurs when occupied automobiles leave.implementing task offloading in a fog computing setting and presenting a GOMOTO approach for task offloading based on the performance model. Multiple Fag Nodes (FNs) working together are assigned tasks to finish.

Jain, V. et al. (2023) used a variety of user quality of service criteria, including end-to-end latency, energy use, task deadline, and priority, to create a Markov Decision Process (MDP) for the task offloading problem. Three separate model-free off-policy Deep Reinforcement Learning (DRL) based approaches are detailed for optimizing the reward in terms of resource utilization.

The goal of the Fog Resource aware Adaptive Task Offloading (FRATO) architecture is to minimize service provisioning latency in IoT-fog-cloud systems (H. Tran-Dang et al., 2021). With the fog resource as its backbone, FRATO makes it easy to choose the optimal offloading strategy on the fly. A data fragment-based collaborative work offloading method is part of this. Additionally, in order to efficiently apply the optimal offloading strategies in the presence of resource competition, two distributed fog resource allocation algorithms, TPRA and MaxRU, are developed.

In 2023, the MISSION method—which stands for Mobility Task Scheduling Offloading—was proposed by Matrouk, K.M. et al. (1). The history-aware handover mechanism regulates the movement of IoT devices to reduce the retransmission rate. With the use of RSS, direction, and distance characteristics, we implemented the Mobility Aware Proximal Policy Optimization (MAPPO) approach to manage the handover. This method is managed by a 5G gateway. (2). Before sending the tasks to the scheduling input, they are sorted into four groups to provide a thorough classification and scheduling process based on several criteria. The scheduling and classification are taken care of using Di-Process Modular Neural Networks (Di-MNN). (3). For energy-efficient work allocation, each job's weight is determined using First Fitness based Animal Migration Optimization (FFAMO). To assign the weighted assignment to the best fog, the capacity-based Hungarian Assignment algorithm (CH2A) is used. One final thing: (4). When making plans or assigning responsibilities When the central fog node is overwhelmed, it generates virtual fog using graph entropy.

task offloading is introduced as a multi-objective optimization problem in Keshavarznejad, M. et al. (2021) that aims to minimize both the total power consumption of the system and the delays in task execution. Next, it uses the Bees algorithm and the non-dominated sorting genetic algorithm (NSGA-II), two meta-heuristic approaches, while considering the NP-hardness of the task.

In their presentation, S. D. Okegbile et al. (2022) outline an edge computing-based system that can handle many users, classes, and layers, allowing for efficient computation and work offloading. A three-layer system is considered, with each device transferring its computed tasks to a different layer at the beginning of each time period. The work is accepted if, upon reaching the selected tier, the queue size is smaller than the predefined threshold; otherwise, it is sent on to the next layer. According to their categories, tasks are grouped in order to meet the quality of service criteria.

Kishor, A. et al. (2022) presented a meta-heuristic scheduler known as Smart Ant Colony Optimization (SACO) task offloading approach for the purpose of offloading the responsibilities of Internet of Things (IoT) sensor applications in a fog environment. There is a comparison of the method's results to those of round robin, modified particle swarm optimization (MPSO), the Bee life algorithm (BLA), and the throttled scheduler algorithm.

An technique based on the postponed acceptance algorithm is proposed by F. Chiti et al. (2018) to provide efficient distributed allocation while assuring stability over the matching result. Using metrics like worst-case total completion time, mean total completion time per work, and mean waiting time, the proposed strategy is tested in computer simulations.

Q. Wu et al. investigate the optimal offloading strategy that accounts for the exit of occupied vehicles (2020). To begin solving the problem of task offloading, a semi-Markov decision process (SMDP) model is required. Next, the value iteration approach for the SMDP was developed with the goal of maximizing the long-term benefit for the vehicular fog and cloud computing system.

M. K. Hussein et al. (2020) present two variants of scheduling techniques based on two natural-source meta-heuristic schedulers, ACO and PSO, to efficiently manage the distribution of Internet of Things (IoT) tasks on fog nodes while containing reaction time and communication costs.

A sub-optimal off-loading technique was provided by D. Wang et al. (2019) in their study on a Gini coefficient-based FCNs selection algorithm (GCFSA). Simultaneously, a distributed resource optimization algorithm (ROAGA) based on evolutionary algorithms is created to tackle the problem of computing resource allocation. The techniques may solve the problem of UE mobility in fog computing networks by drastically decreasing the likelihood of migration.

In their 2019 study, Mukherjee et al. analyze computational offloading for a variety of tasks with different end-user delay constraints. Every task is started by the end users separately. In this case, the user is responsible for transmitting task data to the main fog node, as stated below. The computational capabilities of fog nodes are restricted compared to distant cloud servers, making it impossible to process all task data at the initial fog node within the time delay deadline established by end user applications. Truly, determining the appropriate proportion of task data to be sent to a distant cloud or another local fog node is entirely the responsibility of the primary fog node. It is also important to figure out how many transmission resources should be sent from the fog node to the faraway cloud and how many CPU cycles the fog node needs to process each piece of task data. A quadratically limited quadratic programming solution has been found for the previously discussed problem.

Sun et al. (2020) provide the IoT-fog-cloud architecture as a whole, which makes full use of fog and cloud capabilities. The computation offloading and resource allocation simulations are then based on the energy and time cost minimization challenge. After that, the problem is fixed, energy consumption is improved, and the application request processing time is accelerated using the ETCORA approach.

A. Melbrek et al. (2021) use energy utilization and time delay to send the processed data to decision criteria for work offload of IoT applications. To continue, it presents the race as a game in which the decision-making process of IoT devices on the allocation of tasks is modeled mathematically within the framework of an optimization problem including both energy and latency. Thirdly, it proposes a novel approach to task distribution in which participants learn to anticipate the actions of their fellow competitors. It was also shown that the solution obtained using the proposed technique approaches the Nash equilibrium (NE).

IV. TASK OFFLOADING AND RESOURCE ALLOCATION IN COLLABORATIVE MOBILE EDGE CLOUD COMPUTING

Task offloading for a multiuser and multiserver system is proposed by P. Wang et al. (2022) to assist mobile users with computationally expensive and time-sensitive activities. Each user has their own application that can do a multitude of separate tasks, and all MEC servers are mounted on BSs. Power control, resource allocation, and joint work offloading are all optimized using multiobjective criteria to ensure that user offloading advantages are maximized. This is a mathematical representation of an optimization problem with three objectives, three variables, and multiple goals. A rapid multi-objective evolutionary strategy for decreasing server cost, energy consumption, and response time is presented in this study.

A scenario where MUs might utilize SBS to send their computations to the MEC server is investigated in the 2019 research by Q.-V. Pham et al. As a result of the wire line back connection between the SBS and the macro BS, all offloading MUs share the computing resources. The author suggests breaking the problem down into two parts depending on the specific problem identification: the combined backhaul bandwidth and computing resource allocation subproblem and the offloading choice subproblem. Using a method called JOBCA, two subproblems are addressed sequentially to get a viable solution for the primary issue.

In their comprehensive explanation, Y. Hao et al. (2018) provide a new concept of task caching. Caching completed tasks and their associated data in an edge cloud cache is known as task caching. We next take a look at the issue of shared caching and compute offloading in the context of edge cloud processing and storage constraints. An difficult mixed-integer programming issue is used to represent this subject by the author. Task caching and offloading (TCO) is an innovative technique proposed by this study that use an alternating iterative strategy to solve the problem.

In their description of a task offloading problem, M. Tang et al. (2022) consider the dynamics of the edge load, non-divisible and delay-sensitive activities, and the long-term projected cost. This study presents a distributed strategy based on deep reinforcement learning that does not need models or choices from other devices to determine when to offload. This method enhances the algorithm's predictive power of future costs by combining the LSTM, DQN, and double-DQN techniques.

The goal of offloading, as stated by L. Yang et al. (2018), is to decrease total energy consumption across all system entities while considering compute and service latency limits. This optimization problem involves energy optimization. Now, to solve the problem of energy optimization, propose a strategy based on the artificial fish swarm approach. The scheme's global convergence property is also shown clearly and in detail.Using the idea of software defined networks, M. Chen et al. (2018) build on previous work on the task offloading problem in ultra-dense networks in an effort to decrease latency and preserve the battery life of the user's equipment. to assist mobile users in doing computationally intensive tasks that are time-sensitive. We look at multiobjective optimization for shared work offloading, authority delegation, and resource distribution in order to maximize users' offloading benefits. A multivariable, multiobjective optimization problem is solved by establishing three objectives. An efficient multiobjective evolutionary algorithm decreases response time, limits energy use, and minimizes cost.

A small cell base station (SBS) allows MUs to transfer computations to the MEC server, as stated by Q.-V. Pham et al. (2019). Through a wireless backhaul, the SBS communicates with the macro BS, and the offloading MUs pool their computational capabilities on the MEC server. Two subproblems that the author proposes splitting the issue into are the offloading choice subproblem and the combined backhaul bandwidth and computing resource allocation subproblem. An algorithm known as JOBCA is created to provide a practical solution to the initial problem by iteratively tackling two subproblems.

According to Y. Hao et al. (2018), task caching is a new concept. Using an edge cloud to temporarily store completed jobs and their related data is known as task caching. Later on, we'll take a look at the problem of optimizing task caching and offloading on edge clouds simultaneously while considering processing and storage resource limits. The author poses this difficult question by framing it as mixed integer programming. An effective solution based on an alternating iterative technique, task caching and offloading (TCO), is developed to solve the problem.

Considering non-divisible, delay-sensitive activities and edge load dynamics, M. Tang et al. (2022) create a task offloading problem with the aim of lowering the expected long-term cost. Each device may independently choose how to offload without knowledge of the task models or other devices' offloading decisions, according to the author's proposed model-free deep reinforcement learning-based distributed approach. For a more accurate long-term cost estimate, the method employs the LSTM, dueling deep Q-network (DQN), and double-DQN approaches.

By considering the constraints of processing power and service latency needs, L. Yang et al. (2018) develop an offloading energy optimization problem with the aim of lowering the overall energy consumption at all system entities. The next step in solving the problem of energy optimization is to develop a method that uses an algorithm based on artificial fish swarms. It has also been shown clearly that the method has a global convergence characteristic.

M. Chen et al. (2018) investigate the task offloading problem in ultra-dense networks via the lens of a software-defined network in an effort to decrease latency while maintaining equipment battery life. It gives a detailed mathematical model to translate the problem of job offloading into the NP-hard mixed integer non-linear formula. In order to tackle this optimization problem, it is necessary to break it down into subproblems related to job placement and resource allocation. Once those two subproblems are resolved, we may suggest an offloading strategy that works.

A three-layer system consisting of a device layer, a cloudlet layer, and the cloud itself was suggested by A. Naouri et al. (2021) to address this problem. In order to accomplish computationally intensive tasks efficiently, DCC uses the cloudlet layer and the cloud layer. Hence, DCC may effectively lessen processing time by lowering the amount of data sent to the cloud. This is accomplished by running tasks on the device layer that have a high communication cost but a low compute cost. In an effort to lower task communication costs, the author proposes a greedy task graph partition offloading technique that schedules tasks according to the processing capability of the destination device.

Wireless local area networks (WLANs) and cellular networks are only two examples of the radio access networks that H.-S. Lee et al. (2018) analyze in their study of mobile cloud computing. Other components of this environment include local cloudlets and remote cloud servers. A TMD can transfer jobs to distant cloud servers or cloudlets, however an NTMD can only utilize their cellular data as usual and cannot offload any chores. Particularly, stochastic geometry is used to analyze the outage risk of work offloading in the HMCC system with both cloudlets and distant cloud servers, and the MCC system with just remote cloud servers.

In order to reduce the total time needed to finish the ASA activities and provide a quick service response, X. Li et al. (2023) investigates task offloading for DL-empowered ASA on mobile edge cloud computing networks. In particular, we build an encoder-decoder model based on convolutional neural networks and deploy it at edge servers to extract ASA task properties for better ASA job handling efficiency. On the other hand, edge servers seek to improve UX by identifying the user's tolerance limit using linear regression. In order to address the specified problem, we provide a distributed offloading framework that is both simple and robust, considering particular network constraints such as user association and the storage/processing capacity of edge servers.

Researchers C. Kai et al. (2021) developed a model for collaborative computing that allows mobile devices (MDs) to outsource some of their work to terminals, edge nodes (EN), and cloud centers (CC) via an analysis of the overall offloading computation, computation, and communication resource configuration. Subsequently, introduce the pipeline-based offloading paradigm. This setup enables the MDs and ENs to allocate computation intensive functions to CCs and ENs depending on their communication and computational capacities. While considering the challenging non-convex problems of power allocation, delivery rate, compute resource, and offloading selection, the objective is to minimize the cumulative delay of all MDs, given the pipeline's proposed offloading strategy. Using the following classic sequential convex approximation (SCA) idea, which transforms a non-convex optimization problem into a convex one, may help with the optimization problem.

Considering the dynamics of incoming tasks and the variations in the channel over time, F. Wang et al. (2020) accounts for energy and temporal links for task-to-task causation. The joint-WPT-MEC design ensures the user successfully completes the task while reducing overall transmission energy consumption at the ET over a given finite horizon, subject to the user's optimal task assignment between local computing and offloading and the ET's optimal energy allocation for WPT. The analysis of the first crucial performance restriction begins with the assumption of offline optimization via the use of flawless CSI and TSI. In this case, it finds the optimally organized orientation for solving the energy minimization problem by using convex optimization methods.

X. Huang et al. (2020) presented a quality-of-service (QoS)-aware resource allocation algorithm for reducing network overhead. This algorithm assigns processing and transmission resources based on the relationship of FNs with IDs and the offloading options made. The first step is to develop an AHP-based evaluation model that ranks the importance of various ID tasks and the desired QoS attributes. To follow up, it proposes a system for assigning RBs to IDs taking into account ID priority, satisfaction, and RB quality. Furthermore, in order to enhance FN self-organization and optimize FN-ID matching, a QoS-aware bilateral matching game is created. The offloading decisions are also dependent on the previous steps, which helps to reduce the network load.

V. CHALLENGES AND RESEARCH GABS

Therefore, the goal of the current study was to draw attention to the research limits of task offloading and resource allocation in the MEC environment so that the field's future development might be appropriately handled. The MEC architecture may be defined as one that uses edge servers to provide cloud services close to end users, perhaps enhancing their performance. However, there are still a number of problems, particularly with the two crucial areas of resource use and work offloading. Key research needs in this area are listed below:

1. Managing resources in environments that could be dynamic and heterogeneous is one of the crucial ways used in operations to organize resources efficiently.

• Challenge[Mohammad Yahya Akhlaqi et al. (2023)]: Mobile edge environments are characterized by dynamism by changing network conditions, different user demand and different edge servers. Managing resources given the complex and broad nature of such infrastructure still presents a formidable effort.

• Research Gaps:

o Scalability: Some current resource allocation algorithms cannot efficiently adapt to new large-scale edge nodes or users. New approaches are required for managing extensive network infrastructure with many edge devices.

o Heterogeneity: DLBs may have different capacities and configurations (CPU, memory, storage) and it is imperative to design unified allocation strategies to different type of devices.

o Cross-layer Optimization: Current work describes the need for resource management at the network, computing, as well as the application level to address the issues related to system performance at each level.

Using various energy-efficient offloading techniques to reduce the computational load on portable devices and instead assign them to appropriate, energy-efficient hosts is known as energy-efficient task offloading.

2. According to Muhammad Yahya Akhlaqi et al. (2023), the challenge is: Offloading work to the edge may help alleviate energy-related issues with smartphones and other mobile devices, but because of the always changing environment, it might be challenging to discover the most energy-efficient offloading method.

- Research Gaps:

o Comprehensive Energy Models: There is scant information in the literature and no detailed, precise power models that consider the amount of energy that communication, computation, and overhead for offloading consumes.

o Joint Energy and Performance Optimization: Prior research employs a dual approach but only pays attention to two types of goals – performance and energy. Currently, there are no solutions that make choices that will make the system achieve the best of worst of both worlds, for example, the best worst latency and energy consumption for real-time applications.

o Sustainable Edge Computing: It can be seen that edge data centers or cloudlets as means of improving energy efficiency for the overall network are not as commonly investigated when it comes to techniques like the use of renewable energy or energy harvesting.

3. Latency-Aware Task Offloading

- Challenge [Manzoor Ahmed et al. (2022)]: Latency is one of the major concerns for MEC, but how precisely to control the offloading and local processing of real time tasks is still an open question.

- Research Gaps:

o Predictive Offloading: Despite the work done on predictive models, not much has been done on models that can predict the user mobility, conditions of the network and load of edge server to make ahead of time offloading decisions that can possibly eliminate latencies.

o Fine-Grained Offloading: Offloading implementations of most existing solutions are at a rather coarse granularity: either the whole task or a module. Micro-outsource techniques, where subclip segments of a task are shipped based on latency conditions, are not widely discussed.

o Coordination Between Edge and Cloud: The current scenarios suggest the importance of considering models on how to best interact between the edge and central cloud infrastructure in a way that optimizes for low latency while not burdening edge nodes.

4. User-Centric Offloading and Resource Management

• Challenge [Aisha Muhammad A et al. (2022)]: The requirements that mobile edge environments have to satisfy include, but are not limited to, the application type, QoE, and data privacy. Previous approaches fail to completely estimate the importance of one user to another and the QoE.

• Research Gaps:

o Personalized Offloading Strategies: Existing offloading strategies typically are modeled such that they can be applied to all users, while offloading preferences particular to a user, such as desired latency, or willingness to expend energy, are considered in offloading decisions.

o QoE-Driven Optimization: These considerations define a necessity for the RM frameworks which directly anticipate for the users' QoE, especially in the situations where the users' preferences are contradictory or the available resources at the edge locally are limited.

o Privacy-Aware Task Offloading: Secured data is vulnerable to invasion whenever it is transferred to edge servers. Currently, several heuristics are available but they do not sufficiently explain how to protect users' privacy in the offloading process to achieve performance requirements to the desired level.

5. Security and Trust in Task Offloading

• Challenge [H. Tran-Dang et al. (2022)]: Efficiency and simplicity achieved by pushing task to edge servers brought more instances of them thus making security and trust paramount. However, existing research does not offer a methodology to tackle complex security issues in MEC environments fully.

• Research Gaps:

o Trust Models for Edge Nodes: Thus, to mitigate the risk of delivering tasks to edge nodes since these nodes may possess vulnerabilities different from those inherent in the cloud servers, sound trust and reputation models are required.

o Lightweight Security Mechanisms: Standard protection systems can present vast amounts of overhead and can be all but suitable for the applications where latency is an issue. There is a research need on lightweight security models that provide data integrity and confidentiality while incurring a low performance penalty.

o Data Protection and Privacy Policies: As offloaded tasks towards the edge, there is a lack of research targeting the enforcement of data protection and privacy policies. This also centers on issues such as applying regulatory laws such as GDPR in a decentralized platform.

6. The concept of Mobility Management and Seamless Handover

• Challenge [M. R. Rezaee et al. (2024)]: User mobility, for instance in vehicular networks or mobile applications, becomes an issue as tasks that are assigned to an edge server may have to be transferred to another server.

• Research Gaps:

o Seamless Handover Mechanisms: Current research also does not offer an efficient solution to handover from one edge node to another as users switch networks. This is important specifically for the perspective of continuous service execution in case of such services as video streaming or augmented reality.

o Mobility-Aware Resource Allocation: More is required for algorithms that forecast the users' motion and adapt the resources needed to contribute to the reduction of the influence of the task migration overhead and latency.

o Vertical Handover Between Edge and Cloud: The continuity of service choreography, especially in terms of vertical handover between edge nodes and centralized cloud data centres, remains a research gap.

7. The challenge of joint optimization of networking computation as well and storage resources

• Challenge: In ME environments, joint optimization of networking, computation, and storage resources, yet is difficult, especially under varying scenarios.

• Research Gaps:

o Cross-Resource Optimization: Although there are prior works that focus on resource utilization in either of the three domains – network, computation, or storage, there is a lack of study on how to optimize all the three at the same time.

o Distributed Resource Management: Algorithms for distributed control of resources in distinct multiple edge nodes with an emphasis on overall performance goals remain a topic for research.

8. OFDMA in 5G and Beyond Networks for Offloading Some of the Tasks

• Challenge: With the emergence of 5G and beyond networks, ultra-low latency and high bandwidth are achievable; however, present task offloading approaches may not fully exploit such networks' potential.

• Research Gaps:

o Offloading for 5G Slices: Further work is required to study how Task Offloading can utilize Network Slicing in 5G in which different slices could have different latency and bandwidth profiles.

o Integration with Next-Gen Wireless Networks: Little is known about how other future generations of wireless technologies (e.g., 6G, mmWave) will extend offloading strategies and governing approaches.

The key features used in various research papers are explained in table 2.

VI. CONCLUSION

Although much progress has been made on task offloading and resource allocation in mobile edge cloud environments many issues are still outstanding. Filling these research gaps, especially in dynamic resource management, energy efficiency, user- centric optimization, mobility and security will be important for the success of mobile edge computing in the next generation applications like artificial intelligence services, internet of things, and real time systems.

REFERENCE

1.M. Asim, Y. Wang, K. Wang and P. -Q. Huang, "A Review on Computational Intelligence Techniques in Cloud and Edge Computing," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 6, pp. 742-763, Dec. 2020, doi: 10.1109/TETCI.2020.3007905.

2.R. Jindal, N. Kumar and H. Nirwan, "MTFCT: A task offloading approach for fog computing and cloud computing," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 145-149, doi: 10.1109/Confluence47617.2020.9058209.

3.C. Jiang, X. Cheng, H. Gao, X. Zhou and J. Wan, "Toward Computation Offloading in Edge Computing: A Survey," in IEEE Access, vol. 7, pp. 131543-131558, 2019, doi: 10.1109/ACCESS.2019.2938660.

4.S. Zhang, N. Yi and Y. Ma, "A Survey of Computation Offloading With Task Types," in IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 8, pp. 8313-8333, Aug. 2024, doi: 10.1109/TITS.2024.3410896.

5.M. Aazam, S. u. Islam, S. T. Lone and A. Abbas, "Cloud of Things (CoT): Cloud-Fog-IoT Task Offloading for Sustainable Internet of Things," in IEEE Transactions on Sustainable Computing, vol. 7, no. 1, pp. 87-98, 1 Jan.-March 2022, doi: 10.1109/TSUSC.2020.3028615.

6.Nidhi Kumari, Anirudh Yadav, Prasanta K. Jana, Task offloading in fog computing: A survey of algorithms and optimization techniques, Computer Networks, Volume 214, 2022, 109137, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2022.109137.

7.B. Wang, C. Wang, W. Huang, Y. Song and X. Qin, "A Survey and Taxonomy on Task Offloading for Edge-Cloud Computing," in IEEE Access, vol. 8, pp. 186080-186101, 2020, doi: 10.1109/ACCESS.2020.3029649.

8. B. Kar, W. Yahya, Y. -D. Lin and A. Ali, "Offloading Using Traditional

Optimization and Machine Learning in Federated Cloud–Edge–Fog Systems: A Survey," in IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1199-1226, Secondquarter 2023, doi: 10.1109/COMST.2023.3239579.

9. Yang, Jinming, Awais Aziz Shah, and Dimitrios Pezaros. 2023. "A Survey of Energy Optimization Approaches for Computational Task Offloading and Resource Allocation in MEC Networks" Electronics 12, no. 17: 3548. https://doi.org/10.3390/electronics12173548

10. Akhirul Islam, Arindam Debnath, Manojit Ghose, Suchetana Chakraborty, A Survey on Task Offloading in Multi-access Edge Computing, Journal of Systems Architecture, Volume 118, 2021, 102225, ISSN 1383-7621, https://doi.org/10.1016/j.sysarc.2021.102225

11. Shi Dong, Junxiao Tang, Khushnood Abbas, Ruizhe Hou, Joarder Kamruzzaman, Leszek Rutkowski, Rajkumar Buyya, Task offloading strategies for mobile edge computing: A survey, Computer Networks, Volume 254, 2024, 110791, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2024.110791.

12. M. Mukherjee, S. Kumar, M. Shojafar, Q. Zhang and C. X. Mavromoustakis, "Joint Task Offloading and Resource Allocation for Delay-Sensitive Fog Networks," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-7, doi: 10.1109/ICC.2019.8761239.

Task offloading and resource allocation in edge computing

13. Ullah, I., Lim, HK., Seok, YJ. et al. Optimizing task offloading and resource allocation in edge-cloud networks: a DRL approach. J Cloud Comp 12, 112 (2023). https://doi.org/10.1186/s13677-023-00461-3

14. T. X. Tran and D. Pompili, "Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks," in IEEE Transactions on Vehicular Technology, vol. 68, no. 1, pp. 856-868, Jan. 2019, doi: 10.1109/TVT.2018.2881191.

15. J. Yan, S. Bi, Y. J. Zhang and M. Tao, "Optimal Task Offloading and Resource Allocation in Mobile-Edge Computing With Inter-User Task Dependency," in IEEE Transactions on

Wireless Communications, vol. 19, no. 1, pp. 235-250, Jan. 2020, doi: 10.1109/TWC.2019.2943563.

16. C. -F. Liu, M. Bennis, M. Debbah and H. V. Poor, "Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing," in IEEE Transactions on Communications, vol. 67, no. 6, pp. 4132-4150, June 2019, doi: 10.1109/TCOMM.2019.2898573.

17. H. Jiang, X. Dai, Z. Xiao and A. Iyengar, "Joint Task Offloading and Resource Allocation for Energy-Constrained Mobile Edge Computing," in IEEE Transactions on Mobile Computing, vol. 22, no. 7, pp. 4000-4015, 1 July 2023, doi: 10.1109/TMC.2022.3150432.

18. M. -H. Chen, B. Liang and M. Dong, "Multi-User Multi-Task Offloading and Resource Allocation in Mobile Cloud Systems," in IEEE Transactions on Wireless Communications, vol. 17, no. 10, pp. 6790-6805, Oct. 2018, doi: 10.1109/TWC.2018.2864559.

19. Elgendy, Ibrahim A., Souham Meshoul, and Mohamed Hammad. 2023. "Joint Task Offloading, Resource Allocation, and Load-Balancing Optimization in Multi-UAV-Aided MEC Systems" Applied Sciences 13, no. 4: 2625. https://doi.org/10.3390/app13042625

20. Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu and F. Tian, "Dynamic Task Offloading and Resource Allocation for Mobile-Edge Computing in Dense Cloud RAN," in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3282-3299, April 2020, doi: 10.1109/JIOT.2020.2967502.

21. J. Almutairi, M. Aldossary, H. A. Alharbi, B. A. Yosuf and J. M. H. Elmirghani, "Delay-Optimal Task Offloading for UAV-Enabled Edge-Cloud Computing Systems," in IEEE Access, vol. 10, pp. 51575-51586, 2022, doi: 10.1109/ACCESS.2022.3174127.

22. T. Alfakih, M. M. Hassan, A. Gumaei, C. Savaglio and G. Fortino, "Task Offloading and Resource Allocation for Mobile Edge Computing by Deep Reinforcement Learning Based on SARSA," in IEEE Access, vol. 8, pp. 54074-54084, 2020, doi: 10.1109/ACCESS.2020.2981434.

23. Qian Ren, Kui Liu, Lianming Zhang, Multi-objective optimization for task offloading based on network calculus in fog environments, Digital Communications and

Networks, Volume 8, Issue 5, 2022, Pages 825-833, ISSN 2352-8648, https://doi.org/10.1016/j.dcan.2021.09.012.

24. Jain, V., Kumar, B. QoS-Aware Task Offloading in Fog Environment Using Multi-agent Deep Reinforcement Learning. J Netw Syst Manage 31, 7 (2023). https://doi.org/10.1007/s10922-022-09696-y

25. H. Tran-Dang and D. -S. Kim, "FRATO: Fog Resource Based Adaptive Task Offloading for Delay-Minimizing IoT Service Provisioning," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 10, pp. 2491-2508, 1 Oct. 2021, doi: 10.1109/TPDS.2021.3067654.

26. Matrouk, K.M., Matrouk, A.D. Mobility Aware-Task Scheduling and Virtual Fog for Offloading in IoT-Fog-Cloud Environment. Wireless Pers Commun 130, 801–836 (2023). https://doi.org/10.1007/s11277-023-10310-w

27. Keshavarznejad, M., Rezvani, M.H. & Adabi, S. Delay-aware optimization of energy consumption for task offloading in fog environments using metaheuristic algorithms. Cluster Comput 24, 1825–1853 (2021). https://doi.org/10.1007/s10586-020-03230-y

28. S. D. Okegbile, B. T. Maharaj and A. S. Alfa, "A Multi-User Tasks Offloading Scheme for Integrated Edge-Fog-Cloud Computing Environments," in IEEE Transactions on Vehicular Technology, vol. 71, no. 7, pp. 7487-7502, July 2022, doi: 10.1109/TVT.2022.3167892.

29. Kishor, A., Chakarbarty, C. Task Offloading in Fog Computing for Using Smart Ant Colony Optimization. Wireless Pers Commun 127, 1683–1704 (2022). https://doi.org/10.1007/s11277-021-08714-7

30. F. Chiti, R. Fantacci and B. Picano, "A Matching Theory Framework for Tasks Offloading in Fog Computing for IoT Systems," in IEEE Internet of Things Journal, vol. 5, no. 6, pp. 5089-5096, Dec. 2018, doi: 10.1109/JIOT.2018.2871251.

31. Q. Wu, H. Ge, H. Liu, Q. Fan, Z. Li and Z. Wang, "A Task Offloading Scheme in Vehicular Fog and Cloud Computing System," in IEEE Access, vol. 8, pp. 1173-1184, 2020, doi: 10.1109/ACCESS.2019.2961802.

32. M. K. Hussein and M. H. Mousa, "Efficient Task Offloading for IoT-Based

Applications in Fog Computing Using Ant Colony Optimization," in IEEE Access, vol. 8, pp. 37191-37201, 2020, doi: 10.1109/ACCESS.2020.2975741.

33. D. Wang, Z. Liu, X. Wang and Y. Lan, "Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks," in IEEE Access, vol. 7, pp. 43356-43368, 2019, doi: 10.1109/ACCESS.2019.2908263.

34. M. Mukherjee et al., "Task Data Offloading and Resource Allocation in Fog Computing With Multi-Task Delay Guarantee," in IEEE Access, vol. 7, pp. 152911-152918, 2019, doi: 10.1109/ACCESS.2019.2941741.

35. Sun, H., Yu, H., Fan, G. et al. Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture. Peer-to-Peer Netw. Appl. 13, 548–563 (2020). https://doi.org/10.1007/s12083-019-00783-7

36. A. Mebrek and A. Yassine, "Intelligent Resource Allocation and Task Offloading Model for IoT Applications in Fog Networks: A Game-Theoretic Approach," in IEEE Transactions on Emerging Topics in Computational Intelligence, doi: 10.1109/TETCI.2021.3102214.

37. P. Wang, K. Li, B. Xiao and K. Li, "Multiobjective Optimization for Joint Task Offloading, Power Assignment, and Resource Allocation in Mobile Edge Computing," in IEEE Internet of Things Journal, vol. 9, no. 14, pp. 11737-11748, 15 July15, 2022, doi: 10.1109/JIOT.2021.3132080.

38. Q. -V. Pham, L. B. Le, S. -H. Chung and W. -J. Hwang, "Mobile Edge Computing With Wireless Backhaul: Joint Task Offloading and Resource Allocation," in IEEE Access, vol. 7, pp. 16444-16459, 2019, doi: 10.1109/ACCESS.2018.2883692.

39. Y. Hao, M. Chen, L. Hu, M. S. Hossain and A. Ghoneim, "Energy Efficient Task Caching and Offloading for Mobile Edge Computing," in IEEE Access, vol. 6, pp. 11365-11373, 2018, doi: 10.1109/ACCESS.2018.2805798.

40. M. Tang and V. W. S. Wong, "Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems," in IEEE Transactions on Mobile Computing, vol. 21, no. 6, pp. 1985-1997, 1

June 2022, doi: 10.1109/TMC.2020.3036871.

41. L. Yang, H. Zhang, M. Li, J. Guo and H. Ji, "Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G," in IEEE Transactions on Vehicular Technology, vol. 67, no. 7, pp. 6398-6409, July 2018, doi: 10.1109/TVT.2018.2799620.

42. M. Chen and Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 587-597, March 2018, doi: 10.1109/JSAC.2018.2815360.

43. A. Naouri, H. Wu, N. A. Nouri, S. Dhelim and H. Ning, "A Novel Framework for Mobile-Edge Computing by Optimizing Task Offloading," in IEEE Internet of Things Journal, vol. 8, no. 16, pp. 13065-13076, 15 Aug.15, 2021, doi: 10.1109/JIOT.2021.3064225.

44. H. -S. Lee and J. -W. Lee, "Task Offloading in Heterogeneous Mobile Cloud Computing: Modeling, Analysis, and Cloudlet Deployment," in IEEE Access, vol. 6, pp. 14908-14925, 2018, doi: 10.1109/ACCESS.2018.2812144.

45. X. Li, Z. Xu, F. Fang, Q. Fan, X. Wang and V. C. M. Leung, "Task Offloading for Deep Learning Empowered Automatic Speech Analysis in Mobile Edge-Cloud Computing Networks," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1985-1998, 1 April-June 2023, doi: 10.1109/TCC.2022.3177649.

46. C. Kai, H. Zhou, Y. Yi and W. Huang, "Collaborative Cloud-Edge-End Task Offloading in Mobile-Edge Computing Networks With Limited Communication Capability," in IEEE Transactions on Cognitive Communications and Networking, vol. 7, no. 2, pp. 624-634, June 2021, doi: 10.1109/TCCN.2020.3018159.

47. F. Wang, J. Xu and S. Cui, "Optimal Energy Allocation and Task Offloading Policy for Wireless Powered Mobile Edge Computing Systems," in IEEE Transactions on Wireless Communications, vol. 19, no. 4, pp. 2443-2459, April 2020, doi: 10.1109/TWC.2020.2964765.

48. X. Huang, Y. Cui, Q. Chen and J. Zhang, "Joint Task Offloading and QoS-Aware Resource Allocation in Fog-Enabled

Internet-of-Things Networks," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7194-7206, Aug. 2020, doi: 10.1109/JIOT.2020.2982670.

49. Mohammad Yahya Akhlaqi, Zurina Binti Mohd Hanapi, Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions, Journal of Network and Computer Applications, Volume 212, 2023, 103568, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2022.103568.

50. Abednego Acheampong, Yiwen Zhang, Xiaolong Xu and Daniel Appiah Kumah, A Review of the Current Task Offloading Algorithms, Strategies and Approach in Edge Computing Systems, Computer Modeling in Engineering & Sciences, 2023, vol.134, no.1, DOI: 10.32604/cmes.2022.021394.

51. Manzoor Ahmed, Salman Raza, Muhammad Ayzed Mirza, Abdul Aziz, Manzoor Ahmed Khan, Wali Ullah Khan, Jianbo Li, Zhu Han, A survey on vehicular task offloading: Classification, issues, and challenges, Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 7, 2022, Pages 4135-4162, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2022.05.016.

52. Aisha Muhammad A. Hamdi, Farookh Khadeer Hussain, Omar K. Hussain, Task offloading in vehicular fog computing: State-of-the-art and open issues, Future Generation Computer Systems, Volume 133, 2022, Pages 201-212, ISSN 0167-739X, https://doi.org/10.1016/j.future.2022.03.019.

53. H. Tran-Dang and D. -S. Kim, "A Survey on Matching Theory for Distributed Computation Offloading in IoT-Fog-Cloud Systems: Perspectives and Open Issues," in IEEE Access, vol. 10, pp. 118353-118369, 2022, doi: 10.1109/ACCESS.2022.3219427.

54. M. R. Rezaee, N. A. W. Abdul Hamid, M. Hussin and Z. A. Zukarnain, "Fog Offloading and Task Management in IoT-Fog-Cloud Environment: Review of Algorithms, Networks, and SDN Application," in IEEE Access, vol. 12, pp. 39058-39080, 2024, doi: 10.1109/ACCESS.2024.3375368.

Table 2:

S. NO	AUTHOR	JORNAL	YEAR	TECHNOLOGY	TECHNIQUES	METHODOLOGY	TOOLS	PERFORMANCE METRICS	ALGORITHM	APPROACH
1	Wan Norsyafizan W.Muhammad S.S.Mohf Aris	Engineering Science and Technology	2024	5G Fog Computing	Energy Efficient Task Offloading	Energy Aware Task Offloading	Dev C++	Latency Energy Consumption	Gale Shapley matching algorithm	Bi-Partite Graph Problem A Matching Game
2	H. Tran-Dang	IEEE Access	2022	IoT-Fog-Cloud	Distributed Computation Offloading	Fog Resource aware Adaptive Task Offloading	FRATO OFTOc	Delay Reduction rate Service Request rate Average time for Task Resource Mapping	TPRA MaxRU	Particle Swarm Optimization Approach
3	Amit Kishore Chinmay Chakarbarty	Wireless Personal Communication	2021	Fog computing	Task Offloading	Meta Heuristic	MATLAB	Latency Time IoT Sensors	Smart Ant Colony Optimization Algorithm	IoT Sensor Application based TaskOffloading Method
4	X. Huang	IEEE Internet of Things Journal	2020	Fog-Enabled Internet-of-Things Networks	Joint Task Offloading and QoS-Aware Resource Allocation	an analytic hierarchy process-based evaluation	Simulator	loading balance of the network, improve the RB utilization, and reduce the network overhead.	a resource block (RB) allocation algorithm	a QoS-aware bilateral matching game
5	Mohammad Aazam,Saif ul Islam, Salman	IEEE Transactions on Sustainable Computing	2020	CoT-IoT	Computational Offloading and deep learning	Three tier IOT-Fog-Cloud	SFogSim,AngularJS	Simulation total time,Fog's internal processing, Response	1.GG with FCFS Tuples and Coperation Policy 2. GG with	Random FCFS Static

	Taiq Lone							time	FCFS Tuples and No Coperation Policy 3Global Gateway with Random Tuples and No Cooperation Policy	
6	Qi Zhang, Lin Gui, Fen Hou, Jiacheng Chen, Shichao Zhu, Feng Tian	IEEE Internet of Things Journal	2020	Mobile Edge Computing	Dynamic Task offloading Resource Allocation	Lyapunov optimization	simulation	Energy Efficiency Service Delay	Dynamic Task Offloading and Resource Allocation Algorithm (DTORA) Matching Game-Based Subchannel Allocation Algorithm	An online semidistributed Optimization burst force method Lagrangian dual method
7	F. Wang Jie Xu and Shuguang Cui	IEEE Transactions on Wireless Communications	2020	Wireless Powered Mobile Edge Computing Systems	Optimal Energy Allocation and Task Offloading Policy Partial Offloading Convex Optimization	Non Linear Energy Harvesting Model	Simulation	Average Energy Consumption Static Channel Scenario Time Varying Channel cenario	Two new algorithms for Optimally Solving Problems	Heuristic online Designs
8	C. Kai	IEEE Transactions on Cognitive Communications and Networking	2021	Mobile-Edge Computing Networks	Collaborative Task Offloading	collaborative computing	Monte Carlo simulations	Delivery rate	cloud-edge-end collaborative computing	SCA successive convex approximation
9	X. Li	IEEE Transactions on Cloud Computing	2023	Mobile Edge-Cloud Comput	Task Offloading and Automatic	a low-complexity and distributed	Trace Driven Simulation	Total time Satisfaction rate	NA	convolutional neural network

				ing Networks	Speech Analysis	offloading				based encoder-decoder
10	A. Naouri	IEEE Internet of Things Journal	2021	Mobile-Edge Computing	Optimizing Task Offloading	novel a three-layer task offloading framework named DCC	trace-driven and randomized simulations , Matlab	Communication Cost Average Network Utilization Energy Computation Failed Task Task Computation Delay	Greedy Task Graph Partition Algorithm	Greedy Optimization pproach
11	M. Tang	IEEE Transactions on Mobile Computing	2022	Mobile Edge Comp Uting	Task Offloading	mobile device and edge node models	RMSProp optimizer	(a) ratio of dropped tasks; (b) average delay.	DRL-based Algorithm	DRL Based Approcah
12	P. Wang	IEEE Internet of Things Journal	2022	Mobile Edge Computing	Joint Task Offloading, Power Assignment , and Resource Allocation	multiuser and multiserver scenario	Simulation	Response time Energy Consumption Cost	An efficient multiobjective evolutionary	multivariable and multiobjective optimization
13	M. K. Hussein	IEEE Access	2020	IoT-Based Applications in Fog	Task Offloading	Meta Heuristic	MATLAB	Average response time Iterations Standard Deviation No of IoT Nodes	ACO Task Offloading Algorithm Discrete PSO Algorithm	IoT Fog Model Formulation
14	S. D. Okegbile	IEEE Transactions on Vehicular Technology	2022 1	Edge-Fog-Cloud Computing	Multi-User Tasks Offloading	Multi User Multi Class and Mukti Layer	Monte Corto Simulations SS	Latency Mean Throughput	Algorithm1 :Task Offloading Scheme	Stochastic geometry, Parallel Computing and queueing theory
15	Keshavarznejad, Mohammad Hossein Rezvani	Cluster Computing	2021	fog environments	Delay-aware optimization of energy for task offloading	Heuristic Methodology	iFogSim	Energy Consumption ResponseTime	Non Dominated Sorting Genetic Algorithm Bees Algorithm	Multi Objective Optimization
16	T. Alfakih	IEEE	2020	Mobile	Task	Reinforcem	Simulation	Energy	Reinforcem	Offloadin

		Access		Edge Computing	Offloading and Resource Allocation	ent Learning		Consumption Computing Time Delay	ent Learning-State Action - Reward-State-Action Algorithm	g Decision Based SARSA
17	J. Almutairi	IEEE Access	2022	UAV-Enabled Edge-Cloud Computing	Delay-Optimal Task Offloading	Integer Linear Programming	MATLAB	Energy Consumption Service Time	Delay Optimal Task Offloading Decision	Multi-Tier edge cloud Computing
18	Q. Zhang	IEEE Internet of Things Journal	2020	Mobile-Edge Computing	Dynamic Task Offloading and Resource Allocation	An Online Semi Distributed Optimization	Extensive Simulation	Energy Efficiency Service Delay	Energy Efficiency Service Delay Average Weighted Sum Power	Convex Decomposition method and matching game
19	Ullah, I	J Cloud Comp 12	2023	edge-cloud networks	Optimizing task offloading and resource allocation	Markov Decision Process Deep Reinforcement Learning	PyTorch	Response Rejection rate Task Rejection rate Resource Utilization	Double Deep Q-Network	Double Deep Q-Network Edge Cloud
20	R. Jindal	Data Science & Engineering	2020	fog computing and cloud computing	task offloading	Online Computing Paradigm	Fog Sim Simulator	Execution Time Cost of Execution Energy Requirements	Mobile to Fog and Cloud transfer Algorithm	New Method for task offloading to Fog Nodes and Data Centers
21	J.Liu Quyuan wang	Computer Networks	2023	Mobile Edge Cloud Computing	Optimal Multi user Offloading	Optimal Offloading Selection Strategy based on game model	Numeric Simulation	Minimum Execution total cost	MSOP algorithm MUPC algorithm	Optimal Computation Offloading Policy
22	M.Pendo John Mahenge Chunlin Li	Digital Communication and Networks	2022	Mobile Edge Computing	Energy Efficient Task Offloading	MEC Assisted Model	Simulation	Average Energy Utilization Energy Gain	Energy Efficient Task Offloading Algorithm	Novel Hybrid Approach

	Camilus A.Sanga							Response Delay		
23	J Zhang J Chen Xiang Bao	Network and Computer Application	2022	Cloud Edge Computing	Dependent Task Offloading	Cloud Edge Device Collaborative Offloading Model	Python	Average Completion TIme	Edge Selection Server Algorithm DTO-CED Algorithm	Greedy Policy
24	Alberto Robles-Enciso A.F.Skarmeta	COMPUTERNETWORKS	2023	Edge Computing	Reinforcement Learning Task Offloading	Novel Multilayer Extension (ML-RL)	Pure Edge Sim	Task Success Rate Average Total Time Failed Task due to Latency Average Energy Consumption per device	RL mased Task Offloading Algorithm €-Greedy Multi Layer Q-learning Algorithm	Reinforcement Learning Approach Based on Tabular Learning
25	Minwoo Kim, Jonggyu Jang,Youngchol Choi, and Hyun Jong Yan	Electrical Engineering and Systems Science	2024	Mobile Edge Computing Networks	Distributed Task Offloading and Resource Allocation for Latency Minimization	distributed task offloading framework via UARA optimization under MEC networks	3GPP small cell simulation	Energy Consumption Average Latency	Pricing-Based Distributed Task Offloading Al-gorithm for Latency Minimization	optimization-based approach to minimize delay in multi-cell MEC systems
26	J. Yan	IEEE Transactions on Wireless Communications	2020	Mobile-Edge Computing	Optimal Task Offloading and Resource Allocation	One Cilp Policy	Numerical Simulation	Energy Efficiency Delay	Reduced-Complexity Gibbs Sampling Algorithm	Bi-Section Search method
27	Li Dong , Wenji He and Haipeng Yao	Electronics	2023	Mobile Edge Computing Networks	Task Offloading and Resource Allocation for Tasks with Varied Requirements	OFDMA	Simulation	Delay Processing Time	Linear CD-Aided Optimal Resource Allocation Bisection-Search-Based Resource Allocation	A bisection-search-based resource allocation algorithm combined with a CD-based method
28	Qian Ren	Digital Communic	2022	Fog Environ	task offloading	Network Calculus	JDK12 IntelliJ	Energy Consumpti	Globally Optimal	Equivalent energy

		ations and Networks		ment		Theory	IDEA	on Delay	Multi-Objective Optimization Algorithm for Task Offloading	consumption of FNs
29	Hao Yang, Huifu Zhang, Ziyi Gong	Frontiers in Computing and Intelligent Systems	2023	Mobile Edge Computing-enabled IoT Network	Computation Offloading and Resource Allocation	multiuser and multiserver MEC-enabled IoT network	Extensive simulation	a) Average Delay, (b) Average Energy Consumption	entropy-based deep reinforcement learning for CORA	Entropy-based DRL (EDRL)