Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 2: 2025
ISSN : **1906-9685**

Energy Efficient Task Offloading And Resources Management For Energy Efficient Edge-Cloud Computing Networks

Mr. K.JAYA KRISHNA MCA, M Tech., Shaik. Manisha
#1 Associate Professor Department of Master of Computer Applications
#2 Pursuing M.C.A
QIS COLLEGE OF ENGINEERING & TECHNOLOGY
Vengamukkapalem(V), Ongole, Prakasam dist., Andhra Pradesh- 523272

Abstract: The project aims to enhance task offloading in mobile edge computing by addressing the challenges posed by limited computational resources, focusing on maximizing the number of served mobile devices (MDs) while minimizing energy consumption. To achieve this, two primary optimization problems are formulated: a quantity-driven problem to increase the number of served MDs and an energy-driven problem to minimize energy consumption, both of which are NP-hard mixed-integer nonlinear programming challenges. As a solution, a binary tree-based task offloading (BTTO) scheme is proposed, utilizing convex optimization to efficiently derive optimal task offloading decisions. The algorithm is implemented in a simulation environment, where results demonstrate its effectiveness in serving more devices with reduced energy consumption compared to existing techniques. Additionally, the project incorporates extensions such as data compression to minimize transmission time and energy use, along with security measures using SHA256 hash codes to ensure data integrity during offloading.

*"**Index Terms** - Edge computing, task offloading, resource allocation, energy efficiency".*

1. INTRODUCTION

Mobile devices (MDs) are increasingly used for a wide variety of computationally intensive tasks, but they often face significant limitations in processing power and battery life. This makes it difficult for MDs to handle resource-demanding applications such as real-time data processing, machine learning inference, and complex computations, leading to performance degradation. To overcome these challenges, task offloading to more powerful servers in edge or cloud computing environments has emerged as an effective solution. Task offloading enables MDs to delegate heavy computational tasks to edge or cloud servers that possess superior processing capabilities, thus alleviating the computational burden on the MDs and enhancing overall system performance.

Mobile Edge Computing (MEC), which brings edge servers closer to MDs than traditional cloud infrastructure, offers significant advantages in terms of latency and energy efficiency. By reducing the distance between the MD and the computing resources, MEC minimizes the communication delay that would otherwise occur in conventional cloud-based systems, thereby improving task execution speed and reducing the overall response time for MDs [1]. In edge-cloud collaborative networks, tasks are offloaded from MDs to the most appropriate server (edge or cloud) based on network conditions, resource availability, and task requirements, which enables dynamic and optimal task execution [2][3]. Moreover, such systems often integrate machine learning algorithms to adapt to changing network and computational conditions, improving resource allocation and task offloading efficiency in real time [5].

Efficient task offloading requires effective data transmission between the MD and the selected edge or cloud server. This involves the transfer of task input data to the server, where the task is processed, and the resulting output is sent back to the MD. Data compression techniques are commonly employed to reduce the amount of data transferred, minimizing transmission time and conserving battery life on MDs, which is crucial for maintaining a seamless user experience [4][6]. In addition, the security and privacy of data during this offloading process must be ensured, as the data transmitted between the MD and the server could be vulnerable to attacks. Secure communication protocols and encryption methods play a vital role in protecting the integrity and confidentiality of sensitive information during the offloading process [7][8]. Overall, the seamless integration of task offloading with MEC improves the ability of MDs to handle complex tasks efficiently while addressing the challenges of energy consumption, network latency, and data security.

2. RELATED WORK

Recent advancements in mobile edge computing (MEC) and task offloading strategies have addressed the limitations faced by mobile devices (MDs) in handling resource-intensive tasks. Task offloading, which involves transferring computational tasks from MDs to edge or cloud servers, has emerged as an essential technique to enhance the computational capabilities of MDs, reduce latency, and conserve energy. A key challenge in this area is efficiently managing the offloading process while considering the varying network conditions, task complexity, and resource availability across different computing platforms. Several studies have explored these challenges, proposing various solutions to improve the efficiency and effectiveness of task offloading in mobile edge-cloud environments.

One significant approach for improving task offloading in MEC is through intelligent resource allocation. In the context of post-disaster rescue operations, the need for joint task offloading and resource allocation has been examined, focusing on aerial-terrestrial UAV networks that rely on edge and fog computing [9]. This research highlights the importance of optimal resource distribution in dynamic environments where computing resources are constrained, and task offloading needs to be prioritized based on urgency and resource availability. The study suggests that employing MEC architectures that integrate UAVs and edge computing platforms can improve the overall efficiency of task execution in disaster recovery scenarios by reducing latency and improving the reliability of communication between MDs and servers.

In addition to UAV networks, intelligent task offloading and resource allocation strategies have been applied to vehicular networks. The optimization of task offloading in energy-harvesting UAV systems in low Earth orbit (LEO) is another area of research that emphasizes the use of self-optimization algorithms to enhance task offloading efficiency [10]. The authors propose an intelligent optimization mechanism to manage task offloading in energy-constrained UAVs while maximizing energy harvesting. The approach ensures that tasks are offloaded at optimal times, balancing the energy consumption of the UAVs and the time taken for task execution. This is crucial for maintaining continuous service while operating in environments with fluctuating energy resources.

Another relevant study focuses on multi-objective task offloading for edge-fog-cloud systems, where a firefly algorithm is employed to optimize task offloading decisions based on multiple objectives, such as minimizing latency and energy consumption [11]. This multi-objective optimization approach is particularly effective in scenarios where tasks need to be offloaded to different layers of the computing infrastructure (edge, fog, or cloud) depending on the task's characteristics and the available resources at each layer. The algorithm helps in achieving a balanced trade-off between the task execution time and energy consumption, ensuring the efficient operation of MDs in resource-constrained environments.

In the context of satellite edge computing, deep multi-agent reinforcement learning (MARL) has been explored as a means to optimize task offloading and resource allocation in satellite-based networks [12]. The study presents a novel approach using MARL to enable MDs and satellites to collaboratively optimize the task offloading decision-making process. This research emphasizes the ability of reinforcement learning techniques to adapt to changing network conditions, making it a promising solution for real-time task offloading in large-scale satellite edge networks.

For consumer electronic devices, the use of deep learning-assisted task offloading has gained traction, with studies focusing on multi-modal approaches to enhance task offloading efficiency in IoT-fog architectures [13]. This approach combines various deep learning techniques to learn the most efficient offloading strategies based on data from multiple sources, such as user preferences and task characteristics. By integrating multi-modal deep learning with task offloading, it becomes possible to optimize task distribution and resource usage, ensuring faster and more efficient task completion in consumer electronic environments.

Moreover, there is a growing focus on task offloading strategies that can adapt to the varying characteristics of the network and the tasks themselves. One such strategy is the use of adaptive prioritization in vehicular edge computing systems, where deep reinforcement learning (DRL) algorithms are employed to prioritize tasks based on their urgency and the available resources at the edge [14]. This approach allows for the dynamic adjustment of task priorities, ensuring that critical tasks are offloaded first, thereby reducing latency and improving the overall system's responsiveness.

The concept of hybrid cloud-edge environments for task offloading has also been explored, particularly in the context of vehicular cloud computing. An evolutionary game-theoretic approach to task offloading has been proposed for hybrid vehicular cloud-edge environments [15]. This approach leverages game theory to model the interactions between vehicles, cloud servers, and edge devices, optimizing the decision-making process for task offloading. The study demonstrates the effectiveness of game-theoretic models in capturing the dynamics of collaborative offloading in hybrid cloud-edge systems, where different types of tasks may require different offloading strategies based on their nature and importance.

In multi-UAV systems, joint task offloading and resource allocation have been studied using deep reinforcement learning (DRL) with attention mechanisms. The attention-based DRL approach helps in selecting the most appropriate servers for offloading tasks based on the current network conditions and available resources [16]. This approach enhances the efficiency of task offloading by ensuring that the most critical tasks are offloaded to the servers that can process them most effectively, thereby reducing both latency and energy consumption.

Similarly, a deep reinforcement learning approach has been applied to collaborative task offloading in vehicular edge computing environments, particularly for parking assistance systems. This research explores the integration of DRL with parking assistance services to optimize the offloading of tasks related to vehicle location tracking and navigation [17]. The proposed model improves the system's ability to allocate resources efficiently, minimizing the time required to process vehicle-related tasks while ensuring a seamless experience for users.

Finally, the development of novel task offloading techniques for vehicular edge computing networks has been explored through the use of multi-dimensional models. These models take into account the varying computational, communication, and energy constraints of vehicular systems to propose efficient offloading strategies [18]. The study highlights the importance of considering the multi-dimensional nature of vehicular networks when designing task offloading solutions, as these systems must account for multiple factors, such as vehicle movement, network availability, and task urgency, to ensure optimal performance.

In summary, recent research in task offloading for MEC and edge-cloud systems has explored a wide range of strategies, including reinforcement learning, multi-objective optimization, game theory, and hybrid architectures. These approaches aim to enhance the efficiency of task offloading by reducing latency, conserving energy, and adapting to dynamic network conditions. The integration of intelligent algorithms and optimization techniques is essential for improving the performance of MDs in resource-constrained environments and ensuring seamless task execution in modern computing networks.

3. MATRIALS & METHODS

The proposed system introduces a collaborative edge-cloud computing network aimed at optimizing task offloading for mobile devices (MDs). This framework enables MDs to offload tasks to edge and cloud servers based on task requirements and resource availability, enhancing overall efficiency. The Binary Tree-Based Task Offloading (BTTO) algorithm is employed to dynamically allocate tasks, ensuring optimal energy consumption and maximizing the number of MDs served. By considering real-time network conditions and resource demands, the BTTO algorithm ensures that task offloading decisions are adaptive and performance-driven. This approach reduces latency by utilizing nearby edge resources, while balancing the load on the cloud infrastructure, thus enhancing system resilience. The integration of edge and cloud resources addresses the limitations of traditional centralized computing models, offering a more flexible and scalable solution for task management in dynamic, resource-constrained environments [9][10][11]. This model offers significant improvements in both energy efficiency and responsiveness for mobile computing systems.

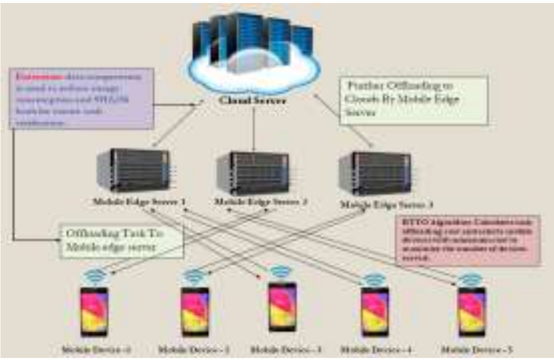


Fig.1 Proposed Architecture

The diagram illustrates a mobile edge computing system designed to optimize task offloading. Mobile devices can offload computationally intensive tasks to nearby mobile edge servers or, if those are overloaded, to a more distant cloud server. The system incorporates multiple edge servers and a central cloud, demonstrating a tiered approach to resource allocation. The "BTTO Algorithm" (likely standing for "Balanced Task-To-Offload") is highlighted as a key component, calculating offloading costs and selecting the optimal server to minimize cost and maximize the number of served devices. Additionally, the diagram mentions data compression and SHA256 hashing for security, suggesting an emphasis on efficient and secure task handling within this distributed computing environment.

i) Generate Edge Network: This step simulates a network of mobile devices (MDs) and edge servers. The user inputs the desired number of MDs, and the system visually represents them as red circles, with edge servers shown as blue circles. The simulation helps set the foundation for the task offloading process, allowing for a visual understanding of the network structure and interactions between MDs and edge servers, which is essential for subsequent task allocation and optimization.

ii) Initialize BTTO Algorithm: This step sets up the Binary Tree-Based Task Offloading (BTTO) algorithm. Each MD is assigned a random offloading cost and energy level, which are crucial for task evaluation. These values allow the BTTO algorithm to make intelligent decisions on task allocation to edge servers, aiming to minimize both offloading costs and energy consumption. The algorithm dynamically optimizes task distribution based on real-time data, ensuring an efficient offloading process across available resources.

iii) Existing Task Offload: This step simulates task offloading using the traditional method, where tasks are allocated to edge servers without considering energy levels or offloading costs. The system tracks energy consumption during this process and measures how many MDs successfully offload their tasks to edge servers. This comparison helps in evaluating the efficiency of the existing method against the proposed BTTO algorithm by quantifying the number of successfully served MDs and the associated energy usage.

iv) Propose Offload Simulation: This step runs the proposed task offloading scheme using the BTTO algorithm. Tasks are optimally assigned from MDs to edge servers, considering both offloading costs and energy consumption. The BTTO algorithm strives to maximize the number of MDs served while minimizing energy consumption. Additionally, SHA256 hash codes are used to verify data integrity throughout the process, ensuring that task data remains consistent and accurate throughout the offloading procedure and does not experience corruption.

v) Served MD Graph: This step generates a graph comparing the number of MDs successfully served by both the existing and proposed task offloading methods. The graph visually displays the improvement in the number of MDs served using the BTTO algorithm. It allows for a clear comparison, showing how the proposed method performs better in terms of serving more MDs than the traditional offloading approach, highlighting the effectiveness of the BTTO algorithm in increasing system efficiency and mobile device handling capacity.

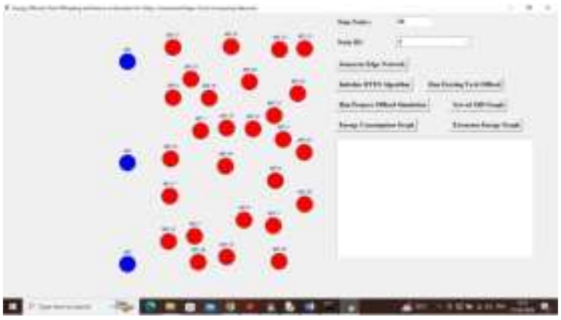
vi) Energy Consumption Graph: This step produces a graph illustrating the energy consumption for both the traditional and BTTO task offloading methods. The graph clearly shows how energy usage differs between the two approaches. By comparing energy consumption, it becomes evident how the BTTO algorithm achieves energy savings, optimizing the allocation of tasks in a way that reduces overall energy expenditure. This visual representation demonstrates the energy efficiency benefits of adopting the BTTO approach over conventional methods.

vii) Extension Energy Graph: This step generates a graph showing energy consumption specifically related to data transmission. It compares energy usage with and without task data compression. The graph demonstrates that using data compression results in lower energy consumption during task transmission, further optimizing the offloading process. By reducing the size of the transmitted data, the energy required for communication is minimized, providing additional efficiency improvements when the BTTO algorithm is combined with data compression techniques.

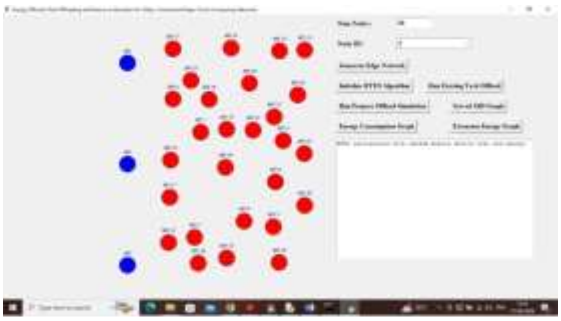
4. RESULTS & DISCUSSION

To run project double click on run.bat file to get below screen

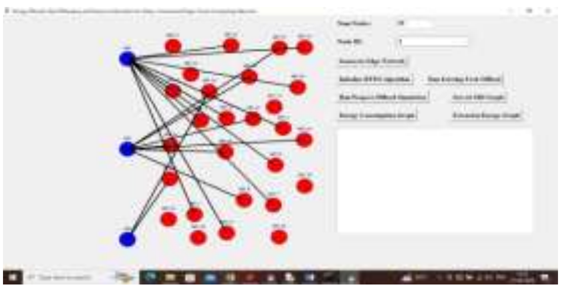
In above screen enter number of nodes in the first text field and then click on ‘Generate Edge Network’ button to get below page



In above screen in entered number of mobile nodes as 30 and after pressing ‘Generate Edge Network’ button will get red colour circles and mobile devices and blue colour circles are the edge servers and now click on ‘Initialize BTTO Algorithm’ button to initialize algorithm with each mobile cost and get below output

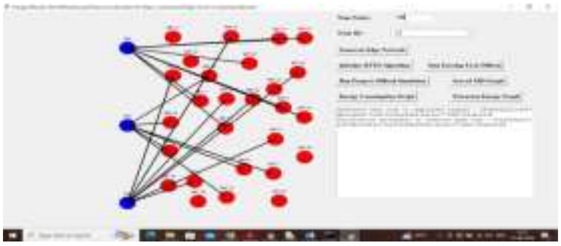


In above screen in text area can see BTTO algorithm initialize with random cost and energy of each mobile node and now click on ‘Run Existing Task Offload’ button to offload task to edge server and get below output

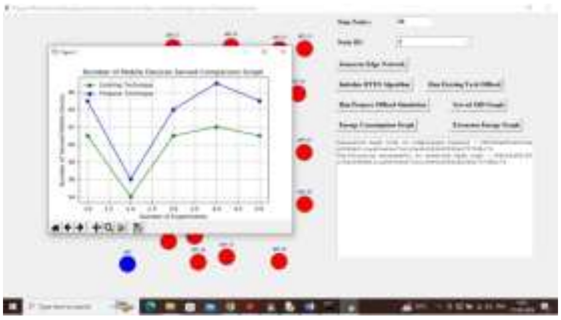


In above screen each node is offloading task to suitable available edge server and each black line indicate task offload between mobile device and Edge Server and now click on ‘Run Propose Offload

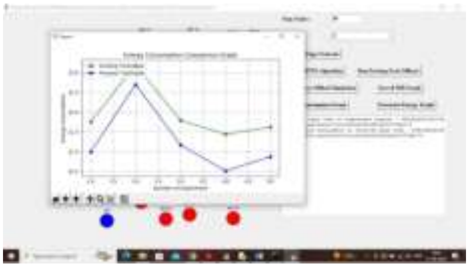
Simulation' button to offload task to edge server by using propose BTTO algorithm and get below page



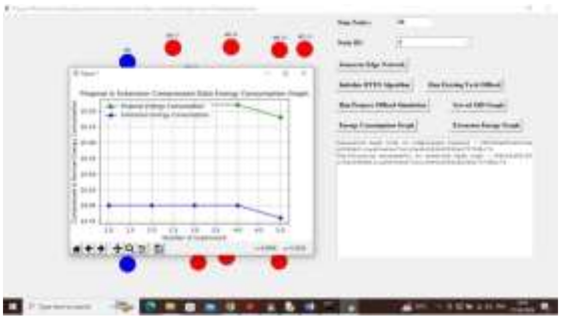
In above screen propose algorithm offloading task to edge server using BTTO algorithm and each request verified using extension SHA256 hash code and in text area can see generated hash code on request before sending and in next line can see hash code after receiving response and both hash code matching so request is successfully verified. By using request authentication extension technique mobile can have secure communication with edge and cloud servers. As extension technique each request data will be compressed to reduce transmission time and energy consumption and run the simulation (for each simulation run click on 'Initialize BTTO Algorithm, Run Existing and Propose buttons) for 4 to 5 times and then generate graphs and then click on "Served MD Graph' button to get below graph



In above graph x-axis represents number of experiments and y-axis represents 'Number of Mobile Devices Served' and green line represents existing techniques and blue line represents propose technique and in both techniques can see Propose algorithm served maximum number of mobile devices and now click on 'Energy Consumption Graph' button to get below graph



In above energy consumption graph x-axis represents number of experiments and y-axis represents energy consumption in joules and then green line represents existing energy consumption and blue line represents propose energy consumption and in both techniques propose got less energy consumption and now click on 'Extension Energy Graph' button to get below extension energy graph



In above extension energy graph x-axis represents 'Number of experiments' and y-axis represents 'Energy Consumption' and then green line represents propose data transfer energy consumption and blue line represents extension compress data transfer energy consumption and in both techniques extension took less energy.

Similarly by following above screens you can run simulation

5. CONCLUSION

The project successfully developed and implemented a task offloading framework that enhances the efficiency of mobile device interactions with edge and cloud servers, leading to more effective processing of long-running tasks. By integrating edge computing with cloud resources, the system improves resource allocation, reducing the burden on any single server and ensuring more balanced and efficient use of available computational power. The project's focus on minimizing energy consumption through optimized task offloading and data compression techniques results in longer battery life for mobile devices and more sustainable operation. The proposed system addresses latency issues by leveraging edge computing to process tasks closer to the user, thus reducing delays and enhancing the responsiveness of the system. The addition of data compression and SHA256 hash code verification ensures that the system not only operates efficiently but also maintains high security and data integrity, safeguarding against potential threats and ensuring reliable task processing.

Future developments of the task offloading framework could focus on several key enhancements to maximize its performance and usability. First, expanding the system to support a larger number of devices and integrating emerging technologies like 5G and IoT would improve its applicability across diverse scenarios. Additionally, implementing enhanced security mechanisms, such as robust encryption and multi-factor authentication, could provide greater protection against evolving cyber threats. Introducing real-time adaptation features that dynamically adjust offloading strategies based on current network conditions and device states could further optimize performance. Improving the user experience by refining the interface to be more intuitive and user-friendly for both developers and end-users is also essential. Finally, integrating AI and machine learning techniques could enhance the Binary Tree-Based Task Offloading (BTTO) algorithm, enabling dynamic adjustments to task offloading decisions through real-time data analysis and predictive modeling, ensuring a more responsive and efficient system.

REFERENCES

[1] S. Wang and Y. Gong, "Joint power control and task offloading in collaborative edge-cloud computing networks", IEEE Internet Things J., vol. 10, no. 17, pp. 15197-15208, Sep. 2023.

[2] W. Chu, P. Yu, Z. Yu, J. C. Lui and Y. Lin, "Online optimal service selection resource allocation and task offloading for multi-access edge computing: A utility-based approach", IEEE Trans. Mobile Comput., vol. 22, no. 7, pp. 4150-4167, Jul. 2023.

[3] Q. Peng, C. Wu, Y. Xia, Y. Ma, X. Wang and N. Jiang, "DoSRA: A decentralized approach to online edge task scheduling and resource allocation", IEEE Internet Things J., vol. 9, no. 6, pp. 4677-4692, Mar. 2022.

[4] L. Tan, Z. Kuang, L. Zhao and A. Liu, "Energy-efficient joint task offloading and resource allocation in OFDMA-based collaborative edge computing", IEEE Trans. Wireless Commun., vol. 21, no. 3, pp. 1960-1972, Mar. 2022.

[5] S. Wang, M. Chen, X. Liu, C. Yin, S. Cui and H. Vincent Poor, "A machine learning approach for task and resource allocation in mobile-edge

computing-based networks", IEEE Internet Things J., vol. 8, no. 3, pp. 1358-1372, Feb. 2021.

[6] Zhu, F., Liu, X., Yu, K., Zhang, Q., Feng, Z., & Li, D. (2024). Delay-effective task offloading technology in Internet of vehicles: from the perspective of the vehicle platooning. *IEEE Transactions on Communications*.

[7] Zhang, C., He, Q., Li, F., & Yu, K. (2024). Intelligent Task Offloading and Resource Allocation in Knowledge Defined Edge Computing Networks. *IEEE Transactions on Mobile Computing*.

[8] Zheng, X., Tahir, M., Aurangzeb, K., Anwar, M. S., Aamir, M., Farzan, A., & Ullah, R. (2024). Non-orthogonal multiple access-based MEC for energy-efficient task offloading in e-commerce systems. *Journal of Cloud Computing*, *13*(1), 117.

[9] Sun, G., He, L., Sun, Z., Wu, Q., Liang, S., Li, J., ... & Leung, V. C. (2024). Joint task offloading and resource allocation in aerial-terrestrial UAV networks with edge and fog computing for post-disaster rescue. *IEEE Transactions on Mobile Computing*.

[10] Alam, A., Shah, P., Trestian, R., Ali, K., & Mapp, G. (2024). Energy Efficiency Optimisation of Joint Computational Task Offloading and Resource Allocation Using Particle Swarm Optimisation Approach in Vehicular Edge Networks. *Sensors*, *24*(10), 3001.

[11] Jia, M., Zhang, L., Wu, J., Guo, Q., Zhang, G., & Gu, X. (2024). Deep Multi-Agent Reinforcement Learning for Task Offloading and Resource Allocation in Satellite Edge Computing. *IEEE Internet of Things Journal*.

[12] Tripathy, S. S., Bebortta, S., ul Haque, M. I., Zhu, Y., & Gadekallu, T. R. (2024). Towards Multi-Modal Deep Learning-Assisted Task Offloading for Consumer Electronic Devices Over An IoT-Fog Architecture. *IEEE Transactions on Consumer Electronics*.

[13] Saif, F. A., Latip, R., Hanapi, Z. M., Shafinah, K., Kumar, A. S., & Bajaher, A. S. (2024). Multi-Objectives Firefly Algorithm for Task Offloading in the Edge-Fog-Cloud Computing. *IEEE Access*.

[14] Lakew, D. S., Tran, A. T., Dao, N. N., & Cho, S. (2024). Intelligent Self-Optimization for Task Offloading in LEO-MEC-Assisted Energy-Harvesting-UAV Systems. *IEEE Transactions on Network Science and Engineering*.

[15] Li, J., Xia, Y., Liu, H., Feng, J., Zhang, K., Ding, Z., ... & Xu, X. (2024, November). An Evolutionary Game Theoretic-Based Approach to Task Offloading in Hybrid Vehicular Cloud-Edge Environment. In *International Conference on Web Services* (pp. 1-15). Cham: Springer Nature Switzerland.

[16] Wu, G., Liu, Z., Fan, M., & Wu, K. (2024). Joint Task Offloading and Resource Allocation in Multi-UAV Multi-Server Systems: An Attention-based Deep Reinforcement Learning Approach. *IEEE Transactions on Vehicular Technology*.

[17] Xue, J., Shao, F., Zhang, T., Tian, G., & Jiang, H. (2024). VEC Collaborative Task Offloading and Resource Allocation Based on Deep Reinforcement Learning Under Parking Assistance. *Wireless Personal Communications*, 1-25.

[18] Abuthahir, S. S., & Peter, J. S. P. (2024). Novel multi-dimensional task offloading techniques for vehicular edge computing networks. *International*

Journal of System Assurance Engineering and Management, 1-14.

[19] Wu, X., Wang, L., Yuan, S., & Chai, W. (2024, June). Task Offloading in Cloud-Edge Collaborative Environment Based on Deep Reinforcement Learning and Fuzzy Logic. In *2024 IEEE 4th International Conference on Software Engineering and Artificial Intelligence (SEAI)* (pp. 301-308). IEEE.

[20] Uddin, A., Sakr, A. H., & Zhang, N. (2024). Adaptive Prioritization and Task Offloading in Vehicular Edge Computing Through Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology*.

Mr. K. Jaya Krishna is an Associate Professor in the Department of Master of Computer Applications at QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He earned his Master of Computer Applications (MCA) from Anna University, Chennai, and his M.Tech in Computer Science and Engineering (CSE) from Jawaharlal Nehru Technological University, Kakinada (JNTUK). With a strong research background, he has authored and co-authored over 90 research papers published in reputed peer-reviewed Scopus-indexed journals. He has also actively presented his work at various national and international conferences, with several of his publications appearing in IEEE-indexed proceedings. His research interests include Machine Learning, Artificial Intelligence, Cloud Computing, and Programming Languages. He is committed to advancing research and fostering innovation while mentoring students to excel in both academic and professional pursuits.

Miss. Shaik.Manisha is a student in the master of computer applications at QIS college of engineering and technology, Ongole, Andhra Pradesh.