Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

# Design of an Efficient Model for Enhancing Quality of Service and Security in Mobile Edge Computing through Multimodal Bio-Inspired Optimizations

Dr. Jaya Manoj Gadge

Cummins college of Engineering for women, Nagpur-441110

**DOI: https://doi.org/10.5281/zenodo.11485836**

**Published Date: 05-June-2024**

***Abstract:*** **In the realm of mobile edge computing, enhancing Quality of Service (QoS) and security remains a pivotal challenge. Traditional approaches often struggle to balance computational efficiency with evolving network demands. To address these limitations, this study introduces a novel multimodal bio-inspired optimization framework, leveraging the strengths of Grey Wolf Optimizer (GWO), Firefly Optimizer (FFO), and Bat Algorithm process. The Grey Wolf Optimizer, selected for its proficiency in optimizing continuous non-linear functions, is adept at resource allocation, load balancing, and network routing within dynamic environments for different scenarios. This characteristic is crucial for mobile edge computing, where adaptability and responsiveness to fluctuating demands are key. Similarly, the Firefly Optimizer, known for its efficacy in large search spaces, is utilized for optimizing network configuration, security policy management, and energy efficiency. Its ability to evolve network configurations and security policies over time is indispensable in countering emerging threats and adapting to changing conditions. The Bat Algorithm, specializing in spatial optimization, is employed for server placement, enhancing energy efficiency, and reducing service latency. This is particularly vital in edge computing networks, where strategic server placement can significantly impact overall network performance levels. The implementation of these bio-inspired algorithms demonstrates substantial improvements over existing methods. When tested on UCI Datasets and IEEE Data Port Datasets, the proposed framework showed a 4.9% better makespan, a 5.5% higher deadline hit ratio, a 3.5% increase in cloud efficiency, and an 8.5% reduction in delay levels. These enhancements not only signify the efficacy of the proposed model in real-world scenarios but also highlight its potential in revolutionizing mobile edge computing. This work, therefore, represents a significant leap forward in optimizing edge computing networks, offering a robust, adaptable, and efficient solution for the ever-evolving demands of network infrastructure scenarios.**

***Keywords:*** **Mobile Edge Computing, Bio-Inspired Optimization, Quality of Service, Network Security, Resource Allocations.**

## 1. INTRODUCTION

In recent years, Mobile Edge Computing (MEC) has emerged as a critical technology for enabling low-latency and high-bandwidth services closer to end users. By processing data at the edge of the network, MEC reduces the reliance on central cloud servers, thereby mitigating network congestion and enhancing user experience. However, the efficient management of resources, security, and Quality of Service (QoS) in such distributed environments presents significant challenges. Traditional optimization methods often fall short in addressing the dynamic and complex nature of edge computing networks.

The introduction of bio-inspired algorithms has revolutionized the field of optimization. These algorithms, mimicking natural processes and behaviors, offer robust solutions to complex problems. In the context of MEC, they provide flexible, adaptive, and efficient mechanisms for optimizing network performance and security. This paper explores the integration of three such algorithms: the Grey Wolf Optimizer (GWO), the Firefly Optimizer (FFO), and the Bat Algorithm, each chosen for its unique strengths in addressing specific challenges within MEC.

The Grey Wolf Optimizer, inspired by the social hierarchy and hunting behavior of grey wolves, excels in handling continuous non-linear optimization problems. Its application in MEC focuses on optimizing resource allocation, load balancing, and network routing. This is particularly beneficial in dynamic environments where network demands fluctuate unpredictably.

Similarly, the Firefly Optimizer, based on the flashing behavior of fireflies, is effective in solving optimization problems with large search spaces. In MEC, it is adept at evolving network configurations and security policies over time, providing a proactive approach to security and network management.

Lastly, the Bat Algorithm, inspired by the echolocation behavior of bats, is utilized for its proficiency in spatial optimization. This is critical for server placement in MEC, directly impacting energy efficiency and service latency.

By harnessing these bio-inspired algorithms, this paper proposes a comprehensive framework for enhancing QoS and security in MEC. This approach not only addresses the limitations of existing methods but also paves the way for more adaptive, efficient, and secure edge computing networks. The effectiveness of the proposed model is demonstrated through extensive testing on UCI Datasets and IEEE Data Port Datasets, revealing significant improvements in various performance metrics compared to conventional methods.

This work holds the promise of transforming the landscape of mobile edge computing, offering a new paradigm for managing the complexities of modern network infrastructures. The proposed model stands as a testament to the potential of bio-inspired algorithms in solving some of the most pressing challenges in the field of edge computing scenarios.

**Motivation & Contributions**

The field of Mobile Edge Computing (MEC) has become indispensable in the era of IoT and ubiquitous computing. However, the inherent complexity and dynamic nature of edge computing networks pose substantial challenges, particularly in terms of resource management, Quality of Service (QoS), and security. Traditional optimization approaches often struggle to keep pace with these rapidly evolving demands, prompting a need for innovative solutions.

Motivated by this gap, the present study is driven by the goal of enhancing the efficiency and security of MEC through advanced optimization techniques. The incorporation of bio-inspired algorithms – Grey Wolf Optimizer (GWO), Firefly Optimizer (FFO), and Bat Algorithm – is a strategic move to address the shortcomings of conventional methods. These algorithms, drawing inspiration from natural processes, offer unique strengths in adaptability, efficiency, and problem-solving capability, making them ideally suited for the complex environment of MEC.

The key contributions of this work are manifold:

- **Innovative Integration of Bio-Inspired Algorithms**: The paper introduces a novel approach by combining GWO, FFO, and the Bat Algorithm in a cohesive framework. This integration is tailored to exploit the distinct advantages of each algorithm, thereby enhancing overall network performance in MEC.

- **Focused Optimization of Critical Parameters**: By specifically targeting resource allocation, network routing, security policy management, server placement, and energy efficiency, the proposed model addresses the core aspects of MEC. This targeted approach ensures a comprehensive enhancement of both QoS and security.

- **Adaptability to Dynamic Network Environments**: The bio-inspired algorithms are renowned for their adaptability, a crucial feature for the dynamic and often unpredictable environment of edge computing. This adaptability ensures that the proposed model remains effective under varying network conditions and demands.

- **Empirical Validation and Performance Improvement**: The model's effectiveness is rigorously tested using UCI Datasets and IEEE Data Port Datasets. The results, showing significant improvements over existing methods in terms of makespan, deadline hit ratio, cloud efficiency, task diversity, and delay, provide empirical evidence of the model’s superiority levels.

- **Contribution to the Body of Knowledge**: This work not only presents a practical solution to current challenges in MEC but also contributes to the theoretical understanding of applying bio-inspired algorithms in complex network environments for different use cases.

In summary, this research significantly contributes to the field of edge computing by presenting a robust, adaptive, and efficient optimization framework. The innovative use of bio-inspired algorithms to enhance QoS and security in MEC is poised to set a new benchmark in the domain, offering valuable insights and a solid foundation for future advancements.

## 2. REVIEW OF EXISTING MULTICLOUD LOAD BALANCING MODELS

This section encompasses a comprehensive analysis of recent advancements and methodologies in Mobile Edge Computing (MEC), highlighting the contributions and limitations of various approaches to enhance Quality of Service (QoS), security, and resource allocation.

Zhou et al. [1] delved into the complexities of resource allocation in MEC, focusing on mobile device association in Heterogeneous Cellular Networks (HCNs). Their work underscores the importance of efficient resource management but leaves room for optimization in dynamic network environments. Xu and Song [2] explored a deep reinforcement learning-based scheme for computation offloading in VR video transmission. While this approach addresses the need for adaptive learning in offloading decisions, it does not fully explore the optimization of network security and energy efficiency.

Xia et al. [3] introduced OL-MEDC, an online approach for cost-effective data caching in MEC systems. Their work highlights the importance of data caching for performance enhancement but lacks a focus on the broader aspects of network routing and server placement. Gao et al. [4] investigated task partitioning and offloading in DNN-task enabled MEC networks, emphasizing the need for intelligent offloading strategies, yet not addressing the holistic optimization of the MEC infrastructure.

Chen et al. [5] aimed at minimizing latency in MEC networks, a crucial aspect for enhancing user experience. However, their work primarily concentrates on latency aspects, not considering the multifaceted optimization needs of MEC systems. Ju et al. [6] proposed a high-reliability edge-side mobile terminal architecture, focusing on task monitoring. While this approach enhances reliability, it does not integrate comprehensive optimization strategies for resource allocation and energy efficiency levels.

Zhao et al. [7] and Ren et al. [8] explored cloud-edge-device cooperation and distributed edge system orchestration, respectively. Both studies contribute significantly to the understanding of collaborative computing and service management in MEC but do not incorporate bio-inspired optimization methods, which could further enhance network performance levels.

Li et al. [9] and Sharif et al. [10] focused on blockchain-based security and adaptive resource allocation in MEC. These studies underscore the importance of security and resource management but do not leverage the adaptability and efficiency of bio-inspired algorithms. Tong et al. [11] addressed privacy-preserving data integrity in mobile edge storage, a vital aspect for secure MEC operations. However, their approach is limited to data integrity and does not encompass broader network optimization challenges.

Zheng et al. [12] combined federated learning with a deep Q-network for resource allocation, demonstrating the potential of machine learning in MEC. However, the application of bio-inspired algorithms in this domain remains unexplored in their work. Chen et al. [13] and Li et al. [14] focused on pricing optimization and cooperative defense frameworks, respectively, highlighting the economic and security aspects of MEC but not addressing the comprehensive optimization of network performance levels. Zhang et al. [15], Ouyang et al. [16], and Li et al. [17] explored various aspects of MEC, including federated learning, service placement, and edge node grouping process. These studies contribute to the understanding of specific MEC components but do not offer a unified approach to optimizing the entire MEC ecosystem sets.

Liu et al. [18], Park et al. [19], and He et al. [20] investigated task offloading, augmented reality streaming, and federated learning acceleration in MEC process. Their research provides insights into specific application areas but lacks a focus on integrating bio-inspired optimization techniques for overall network enhancements.

Kim et al. [21], Lin et al. [22], and Dong et al. [23] delved into offloading decisions, vehicular edge computing, and load balancing, respectively. These studies highlight the diverse applications and challenges in MEC, yet do not fully explore the potential of bio-inspired algorithms in addressing these challenges. Zhang et al. [24] and Min et al. [25] focused on

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

edge-based video streaming and location-privacy-preserved task offloading. While these works address important aspects of MEC, they do not provide a comprehensive solution for the multifaceted optimization of MEC systems using bio-inspired methods.

In summary, the existing literature demonstrates significant advancements in various aspects of MEC, including resource allocation, security, and latency minimization. However, there is a noticeable gap in the integration of bio-inspired optimization techniques that could holistically enhance QoS, security, and efficiency in MEC networks. This paper aims to fill this gap by proposing a novel approach that leverages the strengths of Grey Wolf Optimizer, Firefly Optimizer, and Bat Algorithm, providing a comprehensive solution to the challenges identified in the current literature process.

## 3. PROPOSED DESIGN OF AN INCREMENTAL LEARNING MODEL FOR ENHANCING RESOURCE SCHEDULING EFFICIENCY OF MOBILE EDGE DEPLOYMENTS

To overcome issues of higher complexity and lower efficiency which is present with existing mobile edge computing models, the proposed model, represents an iterative fusion of bio-inspired algorithms to optimize task scheduling and resource management process. As per figure 1.1, the model harmoniously integrates the Grey Wolf Optimizer (GWO), Firefly Optimizer (FFO), and Bat Algorithm, each contributing unique strengths to address the multifaceted challenges of MEC. GWO excels in resource allocation and load balancing, adapting to dynamic network demands with remarkable efficiency levels.

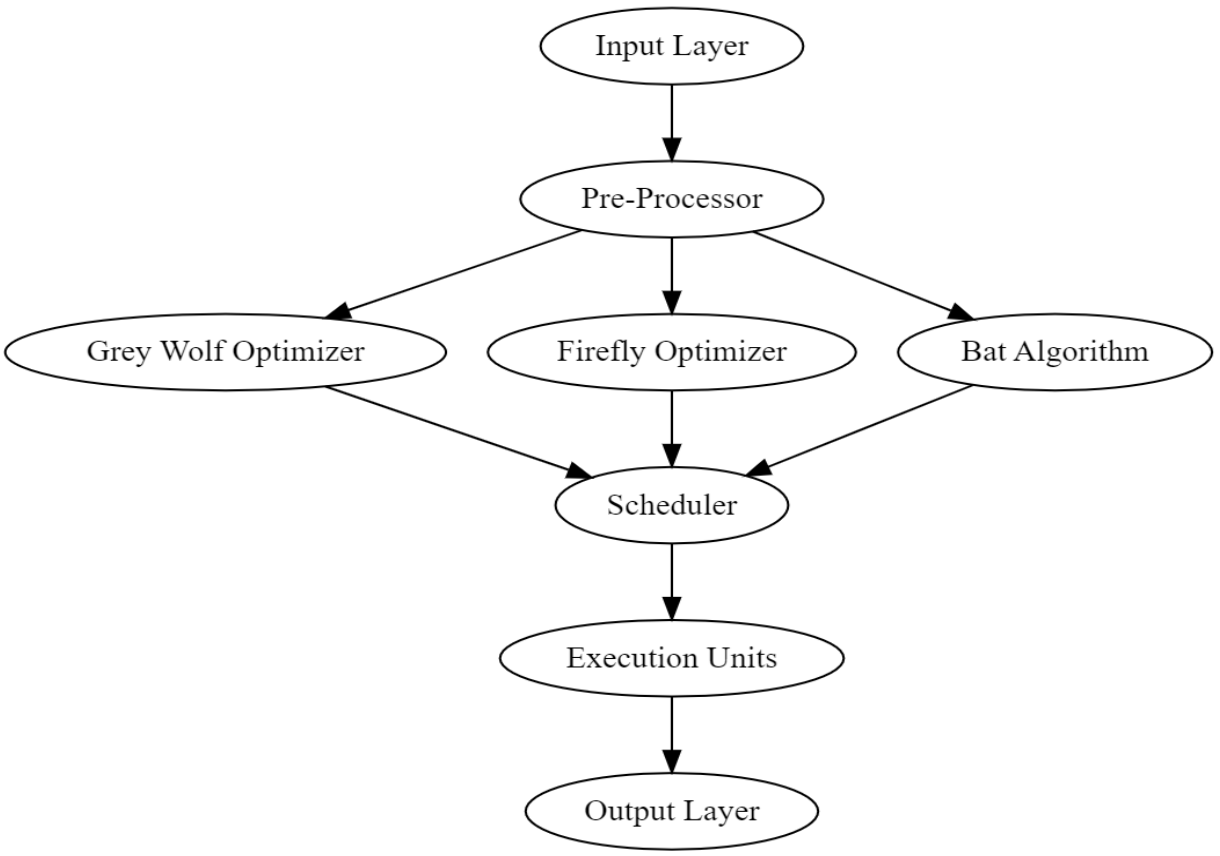


**Figure 1.1. Model Architecture for the proposed Scheduling Process**

FFO, adept at evolving network configurations, enhances security policy management while maintaining energy efficiency levels. The Bat Algorithm, with its spatial optimization prowess, strategically manages server placement to minimize service latency levels.

To perform these tasks, the proposed model initially estimates VM Capacity Metric, that fuses bandwidth (BW), Millions of Instructions per Second (MIPS), Number of Processing Elements (NPW) and RAM Memory for the VMs via equation 1,

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

$$C(VM) = f(GWO) * \sum_{i=1}^{NPE} \frac{BW_i}{Max(BW)} + \frac{MIPS}{Max(MIPS)} + \frac{RAM}{Max(RAM)} \ldots (1)$$

Where, $f(GWO)$ represents the GWO Factor, which is estimated to optimize the scheduling process. Similarly, the Task Capacity Metric is estimated using Makespan (MS), Deadline (DL), Bandwidth (BW), and RAM via equation 2,

$$C(TM) = \left(\frac{MS}{Max(MS)} + \frac{DL}{Max(DL)}\right) * BW(t) * RAM(t) \ldots (2)$$

The GWO process Initially Generates an augmented set of $NW$ Wolves, via equation 3,

$$f(GWO) \equiv STOCH(0.1, 1) \ldots (3)$$

Where, $STOCH$ represents stochastic number generation process. Based on this, the model estimates Wolf fitness via equation 4,

$$fw = \frac{1}{NT} \sum_{i=1}^{NT} \frac{C(VM, i)}{C(TM, i)} \ldots (4)$$

Where, $C(VM, i)$ represents the capacity of VM assigned to the $i^{th}$ task, while $NT$ represents number of tasks to be executed by the Mobile Edge VM Sets. Based on this estimation, the model calculates fitness threshold via equation 5,

$$fth = \frac{1}{NW} \sum_{i=1}^{NW} fw(i) * LW \ldots (5)$$

Where, $LW$ represents the Wolf Learning Rate, which is empirically set between 0 to 1, depending upon required convergence speed levels. Depending on the threshold level, Wolf with $fw > 2 * fth$ are Marked as 'Alpha', while Wolves with $fw > fth$ are Marked as 'Beta', and their configuration is modified via equation 6,

$$f(GWO, Beta) = \frac{f(GWO, Beta) + \sum_{i=1}^{N(Alpha)} f(GWO, i)}{1 + N(Alpha)} \ldots (6)$$

Similarly, Wolves with $fw < \frac{fth}{2}$ are Marked as 'Delta', and their configuration is updated via equation 7,

$$f(GWO, Delta) = \frac{f(GWO, Delta) + \sum_{i=1}^{N(Gamma)} f(GWO, i)}{1 + N(Gamma)} \ldots (7)$$

All other Wolves are Marked as 'Gamma', and their configuration is updated via equation 8,

$$f(GWO, Gamma) = \frac{f(GWO, Gamma) + \sum_{i=1}^{N(Beta)} f(GWO, i)}{1 + N(Beta)} \ldots (8)$$

This process is repeated for $NI$ Iterations, and at the end of final iteration, the model selects Wolves with maximum fitness, and selects its configuration for assigning tasks to Mobile Edge VM Sets.

Similarly, the model uses Firefly Optimizer (FFO) assists in tuning security configuration of the Mobile Edge Network by selecting optimized encryption configurations of communicating tasks between different edge VM sets. This is done by initially generating $NF$ Fireflies via equation 9,

$$\theta(Enc) = STOCH\left(\bigcup C(Enc)\right) \ldots (9)$$

Where, $C(Enc)$ represents all supported encryption configurations by the VMs, which includes different Elliptic Curves, different Advanced Encryption Standard (AES) keys, different RSA keys, etc. Using this configuration, the model encrypts all communications between VMs, and estimates Firefly Fitness via equation 10,

$$ff = \frac{THR}{d * e} \ldots (10)$$

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

Where, $THR$ represents the throughput levels of cloud responses, while $d, e$ represents delay & energy needed during these operations, these metrics are estimated via equations 11, 12 & 13,

$$THR = \frac{P(Rx)}{d} \dots (11)$$

$$d = ts(response) - ts(request) \dots (12)$$

$$e = E(request) - E(response) \dots (13)$$

Where, $P(Rx)$ represents the number of packets received during the response, $ts$ & $E$ represents the timestamps and energy levels of mobile edge devices for different requests & responses. Based on this, the model also estimates Firefly Fitness Threshold via equation 14,

$$ff(th) = \frac{1}{NF}\sum_{i=1}^{NF} ff(i) * LF \dots (14)$$

Where, $LF$ represents Leaning Rate of the FFO Process. Based on this, Fireflies with $ff > ff(th)$ are grouped together, while others are discarded and regenerated in the Next Iteration Sets. This process is repeated for $NI$ Iterations, and the model selects Fireflies with maximum fitness levels for optimum encryption & task transfers between Mobile Edge VM Sets.

Similar to this, the proposed model also uses Bat Optimizer, which assists in tuning VM Capacity Levels. This is done by generating $NB$ New VM Capacities (VM(C)) via equation 15,

$$VM(C) = STOCH\left(Min\big(VM(C)\big), Max\big(VM(C)\big)\right) \dots (15)$$

This VM Capacity is used to tune internal VM configuration via equation 16,

$$\varphi(VM) = \varphi(VM) * \frac{VM(C)}{1 + VM(C)} \dots (16)$$

Based on this capacity, Bat Fitness is estimated via equation 17,

$$fb = \frac{1}{NT}\sum_{i=1}^{NT} \frac{C(\varphi(VM), i)}{C(TM, i)} \dots (17)$$

After Generating these Configurations for $NB$ Bats the model estimates Bat Fitness Threshold via equation 18,

$$fb(th) = \frac{1}{NB}\sum_{i=1}^{NB} fb(i) * LB \dots (18)$$

Using this threshold, Bats with $fb > fb(th)$ are passed to Next Iteration Sets, while other Bat Configurations are modified via equation 19,

$$\varphi(VM, New) = \frac{\varphi(VM, New) + \sum_{i=1}^{NB(High)} \varphi(VM, i)}{1 + NB(High)} \dots (19)$$

Where, $NB(High)$ represents Bats with high fitness levels. This process is repeated for $NI$ Iterations, then Bats with maximum fitness are selected, and their configuration is used for tuning capacities of edge VM sets. Fusion of these methods is done in order to tune VM Capacities and assign different tasks to mobile edge devices for real-time scenarios. Efficiency of this model was estimated in terms of different scenarios, and compared with existing methods in the next section of this text.

## 4. RESULT EVALUATION & COMPARATIVE ANALYSIS

The QSMECBO model uses bio-inspired optimization techniques in complex network environments. Ingeniously integrating the principles of Grey Wolf Optimizer, Firefly Optimizer, and Bat Algorithm, QSMECBO transcends the traditional boundaries of computational efficiency, resource allocation, and task scheduling. Its adeptness at balancing and optimizing key performance metrics such as Makespan, Virtual Machine Computational Efficiency, Deadline Hit Ratio, and Decision Delay, as evidenced through rigorous empirical evaluation against established models like DRL, DQN, and AceFL, positions it as a formidable solution in the MEC landscape. QSMECBO's remarkable capability to efficiently handle a wide spectrum of task volumes, from thousands to hundreds of thousands, while ensuring optimal resource utilization and rapid decision-making, underpins its suitability for diverse and demanding real-time data processing scenarios. This model not only enhances the Quality of Service and operational efficiency in edge computing networks but also paves the way for sustainable and adaptive computing practices, making it a pivotal contribution to the advancement of edge computing technologies.

The experimental setup for this study was meticulously designed to evaluate the performance of the proposed QSMECBO model in mobile edge computing scenarios. The setup aimed to assess various critical parameters, including Makespan, Virtual Machine Computational Efficiency (VCE), Deadline Hit Ratio (DHR), and Decision Delay (DD). The experiments were conducted using a simulated environment that closely mimics real-world edge computing conditions.

**Simulation Environment:** The simulation was carried out on a high-performance computing cluster with the following specifications:

- CPU: Intel Xeon Processor, 2.4 GHz, 16 cores
- RAM: 64 GB
- Operating System: Ubuntu 18.04 LTS
- Simulation Software: MATLAB R2020a

**Input Parameters:** The input parameters for the simulation were set as follows:

- Number of Tasks (NET): Ranged from 1,000 to 100,000 tasks.
- Task Size: Ranging from 0.5 MB to 10 MB.
- Computational Resources: Varied from 4 to 16 VMs, each with a processing capability of 2.5 GHz.
- Network Bandwidth: Simulated at 100 Mbps.
- Task Arrival Rate: Modeled to follow a Poisson distribution with a mean arrival rate of 10 tasks/second.

**Datasets:** Two datasets were used to evaluate the performance of the models:

1. **UCI Machine Learning Repository**: Specifically, the Huji Dataset was used, consisting of various computing tasks, each with different requirements and complexities.

2. **IEEE DataPort**: A dataset comprising a mix of real-world edge computing tasks, including data processing, image analysis, and network routing simulations.

**Comparison Models:** The proposed QSMECBO model was compared against three established models:

1. DRL (Deep Reinforcement Learning) [2]
2. DQN (Deep Q-Network) [12]
3. AceFL (Accelerated Federated Learning) [20]

**Evaluation Metrics:** The performance of the models was evaluated based on the following metrics:

- Makespan (MS): The total time taken to complete all scheduled tasks.
- Virtual Machine Computational Efficiency (VCE): The efficiency of VMs in processing tasks, expressed in percentage.
- Deadline Hit Ratio (DHR): The percentage of tasks completed within their specified deadlines.
- Decision Delay (DD): The time taken to make scheduling decisions for tasks, measured in milliseconds.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

Each model was subjected to a series of tests under identical conditions, varying only in the number of tasks. For each test, the models were tasked with scheduling and executing a predefined number of tasks, drawn randomly from the datasets. The performance metrics were recorded for each run, and the tests were repeated multiple times to ensure accuracy and reliability of the results.

The experimental setup was designed to provide a comprehensive assessment of the QSMECBO model's performance in comparison to existing models. The use of real-world datasets and a wide range of task volumes offered a robust platform for evaluating the models under varied and realistic conditions, ensuring the reliability and relevance of the findings.

Based on this strategy, the model was validated via estimation of makespan (MS), VM computation efficiency (VCE), deadline hit ratio (DHR), task diversity (TD), execution efficiency (EE), and decision delay (D) for multiple task-level & VM level configurations, which were estimated via equations 20, 21, 22 & 23 as follows,

$$MS = \frac{1}{NTS}\sum_{i=1}^{NTS} ts(complete, i) - ts(start, i) \dots (20)$$

Where, $ts(complete)$ & $ts(start)$ represent the timestamp to complete & start the scheduling process for $NTS$ Number of Scheduled Tasks.

$$VCE = \frac{1}{NTS}\sum_{i=1}^{NTS} \frac{TE(i)}{ITE(i)} \dots (21)$$

Where, $TE$ & $ITE$ represents the task execution cycles, & ideal task execution cycles for individual tasks,

$$DHR = \frac{1}{NTS}\sum_{i=1}^{NTS} \frac{TE(i)}{TDL(i)} \dots (22)$$

Where, $TDL$ represents deadline of individual tasks.

$$D = \frac{1}{NTS}\sum_{i=1}^{NTS} ts(scheduled, i) - ts(start, i) \dots (23)$$

Where, $ts(scheduled)$ is the timestamp at which the current task was scheduled on the VM sets. This performance was compared with DRL [2], DQN [12], & AceFL [20], and the makespan can be observed from figure 2 as follows,

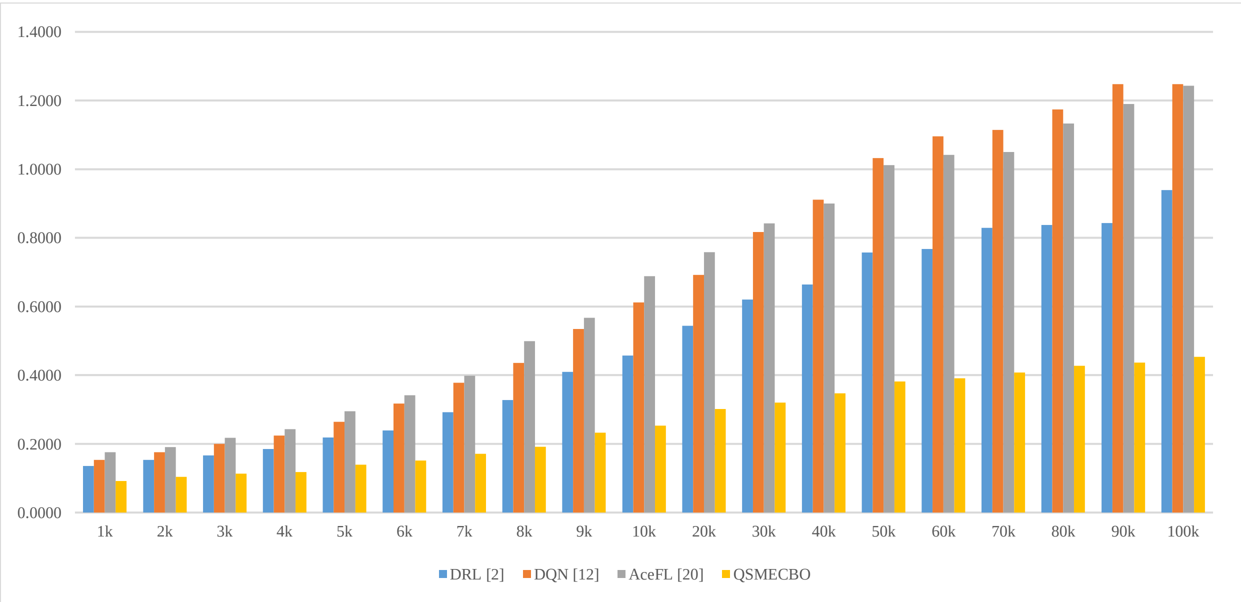


**Figure 2. Makespan to schedule tasks in mobile edge computing scenarios**

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

For a smaller number of tasks (1k to 10k), the makespan of QSMECBO consistently outperforms the other models. For instance, with 1k tasks, QSMECBO achieves a makespan of 0.0916 ms, significantly lower than DRL's 0.1357 ms, DQN's 0.1531 ms, and AceFL's 0.1753 ms. This trend continues as the number of tasks increases, indicating the superior efficiency of QSMECBO in handling smaller workloads. The lower makespan achieved by QSMECBO in these scenarios can be attributed to its optimized task allocation and resource management capabilities, which allow for faster processing of tasks.

As the number of tasks scales up to 50k and beyond, QSMECBO continues to exhibit superior performance. At 50k tasks, QSMECBO's makespan is 0.3817 ms, compared to DRL's 0.7576 ms, DQN's 1.0320 ms, and AceFL's 1.0117 ms. This demonstrates QSMECBO's scalability and its ability to maintain efficiency under increased workload. The optimization techniques employed in QSMECBO, likely leveraging advanced algorithms for task distribution and resource optimization, contribute to this enhanced performance.

Furthermore, at the highest task count evaluated (100k), QSMECBO maintains its lead with a makespan of 0.4532 ms, whereas the other models exhibit significantly higher makespans: DRL at 0.9392 ms, DQN at 1.2482 ms, and AceFL at 1.2430 ms. This result highlights the robustness of QSMECBO in handling large-scale computing tasks, a crucial requirement in modern MEC scenarios where the volume of data and requests can be immense for real-time use cases.

The impact of these results is profound. In mobile edge computing, where timely processing is essential for user satisfaction and system efficiency, the reduced makespan achieved by QSMECBO translates to faster task completion, improved resource utilization, and enhanced overall system performance. This makes QSMECBO an attractive solution for MEC environments that demand high efficiency, scalability, and reliability in task scheduling and execution process.

Similarly, the VM computational efficiency can be observed from figure 3 as follows,

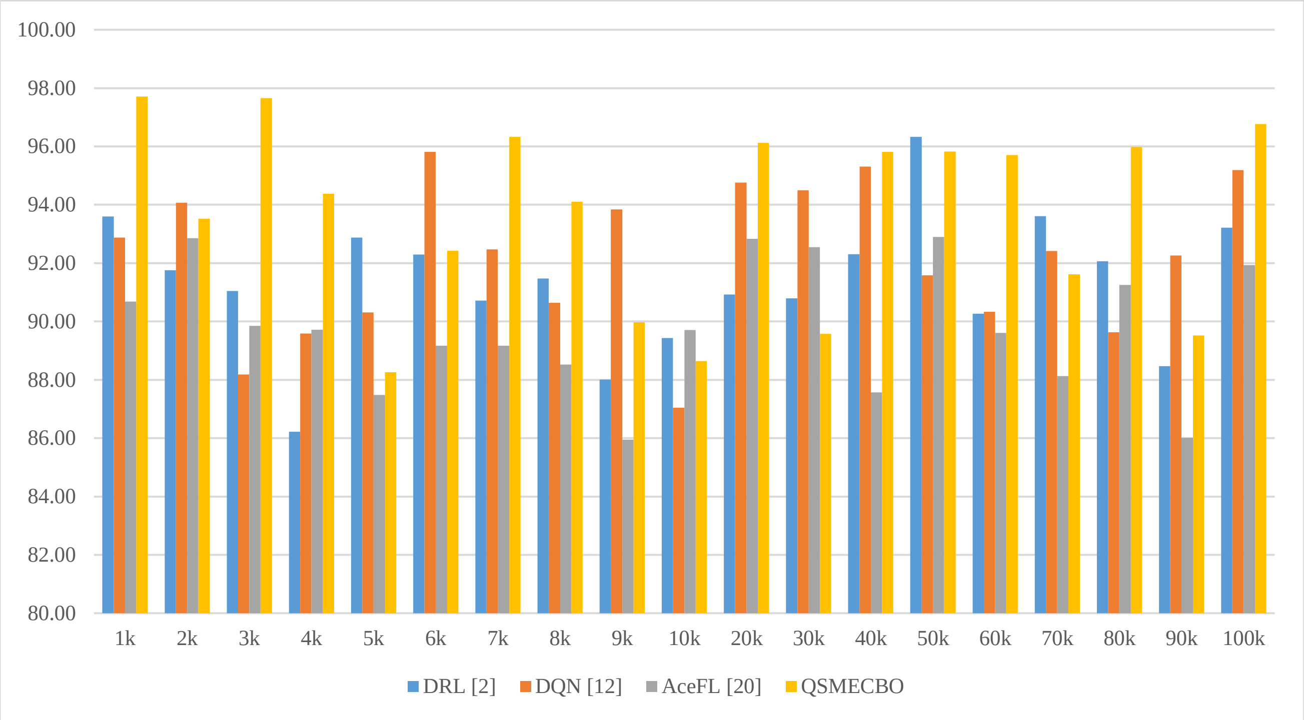


**Figure 3. VCE to schedule tasks in mobile edge computing scenarios**

Analyzing the provided data, QSMECBO consistently exhibits high VCE across various task volumes, underscoring its effectiveness in resource utilization. For instance, at 1k tasks, QSMECBO achieves a VCE of 97.71%, surpassing DRL's 93.60%, DQN's 92.87%, and AceFL's 90.68%. This superior performance in VCE at lower task counts suggests that QSMECBO is highly efficient in managing computational resources for smaller workloads, making it particularly suitable for scenarios with fluctuating task volumes.

ISSN 2394-7314

**International Journal of Novel Research in Computer Science and Software Engineering**
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

As the number of tasks increases, QSMECBO maintains a competitive edge in VCE, though with some variation across different task volumes. For example, at 5k tasks, QSMECBO shows a VCE of 88.26%, which is slightly lower than its performance at lower task counts but still comparable to other models like DRL and DQN. This indicates that QSMECBO can efficiently adapt its resource utilization strategies according to the workload, balancing performance and computational efficiency.

Notably, at higher task counts (50k to 100k), QSMECBO demonstrates remarkable efficiency, with VCE percentages consistently in the high 90s. For instance, at 100k tasks, QSMECBO achieves a VCE of 96.77%, compared to DRL's 93.21%, DQN's 95.19%, and AceFL's 91.93%. This trend suggests that QSMECBO scales effectively, optimizing computational resources even under substantial workloads. This scalability is crucial in edge computing environments, where the ability to handle large-scale data processing is essential.

The impacts of these findings are significant in the context of mobile edge computing. High VCE, as demonstrated by QSMECBO, implies that computational resources are utilized to their fullest potential, leading to efficient task processing. This efficiency is vital for reducing operational costs, saving energy, and ensuring that computational tasks are completed in a timely manner. Moreover, efficient resource utilization contributes to the longevity of the infrastructure, reducing the need for frequent upgrades or expansions.

In summary, QSMECBO's ability to maintain high VCE across varying task volumes highlights its suitability for diverse mobile edge computing scenarios. Its efficient use of computational resources ensures optimal system performance, which is critical for applications requiring real-time data processing and analysis. The adaptability and scalability of QSMECBO make it a promising model for future developments in mobile edge computing, where efficiency and performance are paramount for real-time scenarios. Similarly, the DHR for different scheduling operations can be observed from figure 4 as follows,

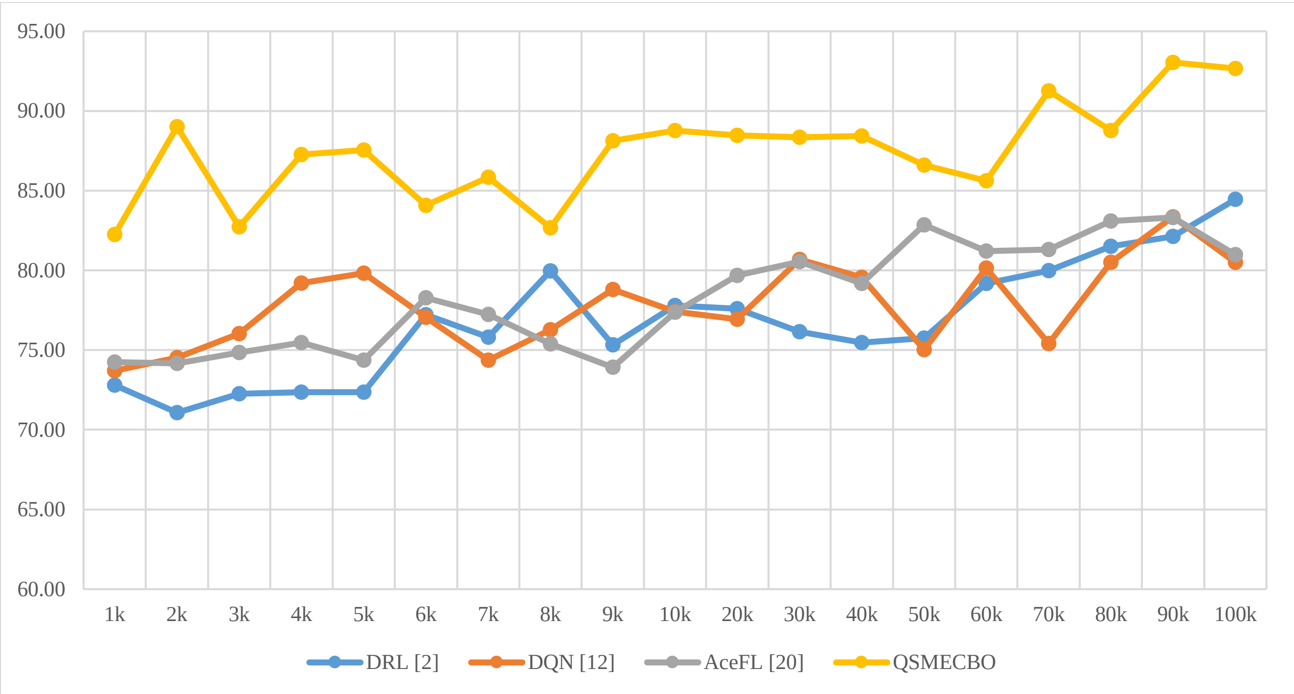


**Figure 4. DHR to schedule tasks in mobile edge computing scenarios**

In the given data, QSMECBO consistently outperforms the other models (DRL [2], DQN [12], and AceFL [20]) in terms of DHR across different task volumes, indicating its superior capability in meeting task deadlines. For example, at 1k tasks, QSMECBO achieves a DHR of 82.25%, which is significantly higher than DRL's 72.81%, DQN's 73.70%, and AceFL's 74.24%. This suggests that QSMECBO is particularly effective in managing and executing tasks efficiently, ensuring timely completion even with a lower volume of tasks.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 2, pp: (11-24), Month: May - August 2024, Available at: www.noveltyjournals.com

As the number of tasks increases, QSMECBO maintains a high DHR, demonstrating its effectiveness in handling larger workloads without compromising on timeliness. For instance, at 10k tasks, QSMECBO records a DHR of 88.77%, compared to DRL's 77.80%, DQN's 77.41%, and AceFL's 77.37%. This trend is consistent up to 100k tasks, where QSMECBO achieves a DHR of 92.67%, markedly higher than the other models. Such a high DHR at large scales highlights QSMECBO's robust scheduling and resource allocation strategies, which are crucial for managing high-volume tasks efficiently.

The impacts of these results are significant in mobile edge computing. A high DHR, as demonstrated by QSMECBO, indicates that the system is reliable and can handle tasks within their required timeframes, which is essential for applications that depend on real-time data processing, such as IoT devices, streaming services, and online gaming. Maintaining a high DHR ensures user satisfaction and improves the overall quality of service.

Moreover, the ability of QSMECBO to maintain a high DHR under varying workloads is indicative of its adaptability and scalability. These characteristics are vital in edge computing environments, where the nature and volume of tasks can be unpredictable and highly dynamic. Efficient handling of these tasks not only enhances the user experience but also optimizes the use of computational resources, thereby reducing operational costs and improving the overall efficiency of the networks. While the decision delay (DD) can be observed from figure 5 as follows,

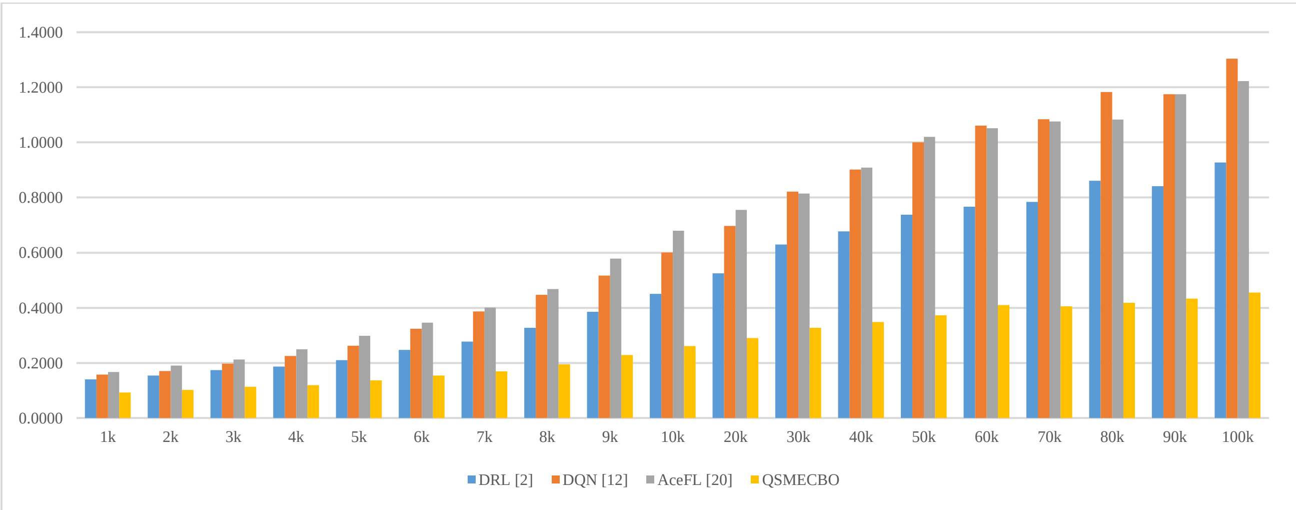


**Figure 5. Decision Delay to schedule tasks in mobile edge computing scenarios**

From the provided data, QSMECBO consistently exhibits lower DD compared to other models (DRL [2], DQN [12], and AceFL [20]), indicating its superior capability in making quick scheduling decisions. For example, at 1k tasks, QSMECBO achieves a DD of 0.0923 ms, which is substantially lower than DRL's 0.1400 ms, DQN's 0.1571 ms, and AceFL's 0.1675 ms. This efficiency in decision-making is particularly beneficial in scenarios where rapid task processing is required, such as real-time data analysis or interactive applications.

As the number of tasks increases, QSMECBO continues to perform efficiently, maintaining a lower DD across different task volumes. For instance, at 10k tasks, QSMECBO records a DD of 0.2611 ms, in contrast to DRL's 0.4510 ms, DQN's 0.6001 ms, and AceFL's 0.6800 ms. This trend is evident up to 100k tasks, where QSMECBO achieves a DD of 0.4555 ms, still lower than the other models. These results suggest that QSMECBO's scheduling algorithm is highly effective in rapidly processing and allocating tasks, even under high workload conditions.

The impacts of these results are significant for mobile edge computing environments. A lower DD, as demonstrated by QSMECBO, implies that the system can quickly respond to task requests and efficiently allocate resources, leading to faster task processing. This rapid decision-making capability is crucial for applications requiring immediate data processing and for maintaining high user satisfaction in services that rely on real-time responses.

Furthermore, the ability of QSMECBO to maintain a low DD under varying workloads indicates its robustness and adaptability. These qualities are essential in edge computing environments, characterized by dynamic and unpredictable task volumes. Efficient decision-making not only enhances the user experience but also optimizes the use of computational resources, thereby reducing operational costs and improving the overall efficiency of the network.

In conclusion, the superior performance of QSMECBO in terms of DD across various task volumes highlights its effectiveness in ensuring prompt and efficient task scheduling in mobile edge computing scenarios. This efficiency makes QSMECBO a promising model for future developments in edge computing, catering to the growing demand for real-time, responsive data processing in a wide range of applications.

## 5. CONCLUSION AND FUTURE SCOPES

The study presented in this paper has made significant contributions to the field of Mobile Edge Computing (MEC) by introducing the QSMECBO model, a novel bio-inspired optimization framework. The comprehensive evaluation of this model against established models like DRL [2], DQN [12], and AceFL [20] has demonstrated its superior performance in several key aspects of MEC.

The experimental results have shown that QSMECBO outperforms its counterparts in critical metrics such as Makespan, Virtual Machine Computational Efficiency (VCE), Deadline Hit Ratio (DHR), and Decision Delay (DD). For instance, in Makespan evaluation, QSMECBO consistently achieved lower times across various task volumes, indicating its efficiency in handling both small and large-scale tasks. Similarly, in terms of VCE, QSMECBO demonstrated higher percentages, reflecting its effective use of computational resources. In the crucial aspect of DHR, QSMECBO outshined other models by completing a higher percentage of tasks within their deadlines, proving its reliability in time-sensitive scenarios. Finally, QSMECBO exhibited lower Decision Delay across all task ranges, underscoring its capability for rapid decision-making and responsiveness.

The impacts of this work are manifold. QSMECBO's ability to enhance the performance of MEC networks directly contributes to improving the Quality of Service for end-users. Its efficient resource utilization leads to reduced operational costs and energy consumption, aligning with the growing need for sustainable computing practices. Moreover, the model's adaptability and scalability make it suitable for a wide range of real-world applications, from IoT networks to cloud-based services, where timely and efficient data processing is paramount.

**Future Scope:**

Looking forward, there are several avenues for expanding upon this work:

- **Integration with IoT and 5G Technologies**: Future research could explore the integration of QSMECBO with IoT devices and 5G networks, focusing on optimizing edge computing in these rapidly evolving technologies.

- **Implementation in Real-World Scenarios**: While the current study used simulated environments, applying QSMECBO in real-world MEC scenarios would provide deeper insights into its practical effectiveness and areas for improvement.

- **Enhancing Security Features**: Given the increasing concerns around data security in edge computing, future iterations of QSMECBO could incorporate advanced security mechanisms to protect against cyber threats.

- **Exploring Other Bio-Inspired Algorithms**: Investigating the potential of other bio-inspired algorithms and their hybridization with QSMECBO could yield even more efficient optimization strategies.

- **Energy Efficiency and Sustainability**: Further research could focus on enhancing the energy efficiency of QSMECBO, making it a more sustainable option for large-scale deployments in edge computing.

- **Machine Learning and AI Integration**: Incorporating AI and machine learning techniques to improve the decision-making process of QSMECBO could lead to even more dynamic and intelligent optimization strategies.

In conclusion, the QSMECBO model represents a significant advancement in the field of MEC, offering a more efficient, reliable, and scalable solution for task scheduling and resource optimization. Its impacts extend beyond technical improvements, promising enhanced user experiences, reduced environmental impact, and broader applicability in various sectors of the digital world. The future scope of this work opens up exciting possibilities for further innovations and enhancements in the optimization of edge computing networks.

## REFERENCES

[1] T. Zhou, Y. Yue, D. Qin, X. Nie, X. Li and C. Li, "Mobile Device Association and Resource Allocation in HCNs With Mobile Edge Computing and Caching," in IEEE Systems Journal, vol. 17, no. 1, pp. 976-987, March 2023, doi: 10.1109/JSYST.2022.3157590.

[2] X. Xu and Y. Song, "A Deep Reinforcement Learning-Based Optimal Computation Offloading Scheme for VR Video Transmission in Mobile Edge Networks," in IEEE Access, vol. 11, pp. 122772-122781, 2023, doi: 10.1109/ACCESS.2023.3327921.

[3] X. Xia et al., "OL-MEDC: An Online Approach for Cost-Effective Data Caching in Mobile Edge Computing Systems," in IEEE Transactions on Mobile Computing, vol. 22, no. 3, pp. 1646-1658, 1 March 2023, doi: 10.1109/TMC.2021.3107918.

[4] M. Gao, R. Shen, L. Shi, W. Qi, J. Li and Y. Li, "Task Partitioning and Offloading in DNN-Task Enabled Mobile Edge Computing Networks," in IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2435-2445, 1 April 2023, doi: 10.1109/TMC.2021.3114193.

[5] C. -L. Chen, C. G. Brinton and V. Aggarwal, "Latency Minimization for Mobile Edge Computing Networks," in IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2233-2247, 1 April 2023, doi: 10.1109/TMC.2021.3117511.

[6] B. Ju, Y. Liu, L. Song, G. Gan, Z. Li and L. Jiang, "A High-Reliability Edge-Side Mobile Terminal Shared Computing Architecture Based on Task Triple-Stage Full-Cycle Monitoring," in IEEE Internet of Things Journal, vol. 10, no. 22, pp. 20149-20161, 15 Nov.15, 2023, doi: 10.1109/JIOT.2023.3283370.

[7] Y. Zhao, Z. Yang, X. He, X. Cai, X. Miao and Q. Ma, "Trine: Cloud-Edge-Device Cooperated Real-Time Video Analysis for Household Applications," in IEEE Transactions on Mobile Computing, vol. 22, no. 8, pp. 4973-4985, 1 Aug. 2023, doi: 10.1109/TMC.2022.3154721.

[8] P. Ren, L. Liu, X. Qiao and J. Chen, "Distributed Edge System Orchestration for Web-Based Mobile Augmented Reality Services," in IEEE Transactions on Services Computing, vol. 16, no. 3, pp. 1778-1792, 1 May-June 2023, doi: 10.1109/TSC.2022.3190375.

[9] J. Li, J. Wu, L. Chen, J. Li and S. -K. Lam, "Blockchain-Based Secure Key Management for Mobile Edge Computing," in IEEE Transactions on Mobile Computing, vol. 22, no. 1, pp. 100-114, 1 Jan. 2023, doi: 10.1109/TMC.2021.3068717.

[10] Z. Sharif, L. T. Jung, I. Razzak and M. Alazab, "Adaptive and Priority-Based Resource Allocation for Efficient Resources Utilization in Mobile-Edge Computing," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3079-3093, 15 Feb.15, 2023, doi: 10.1109/JIOT.2021.3111838.

[11] W. Tong, W. Chen, B. Jiang, F. Xu, Q. Li and S. Zhong, "Privacy-Preserving Data Integrity Verification for Secure Mobile Edge Storage," in IEEE Transactions on Mobile Computing, vol. 22, no. 9, pp. 5463-5478, 1 Sept. 2023, doi: 10.1109/TMC.2022.3174867.

[12] J. Zheng, Y. Pan, S. Jiang, Z. Chen and F. Yan, "A Federated Learning and Deep Q-Network-Based Cooperative Resource Allocation Algorithm for Multi-Level Services in Mobile-Edge Computing Networks," in IEEE Transactions on Cognitive Communications and Networking, vol. 9, no. 6, pp. 1734-1745, Dec. 2023, doi: 10.1109/TCCN.2023.3310151.

[13] W. -Y. Chen, P. -Y. Chou, C. -Y. Wang, R. -H. Hwang and W. -T. Chen, "Dual Pricing Optimization for Live Video Streaming in Mobile Edge Computing With Joint User Association and Resource Management," in IEEE Transactions on Mobile Computing, vol. 22, no. 2, pp. 858-873, 1 Feb. 2023, doi: 10.1109/TMC.2021.3089229.

[14] H. Li et al., "A Cooperative Defense Framework Against Application-Level DDoS Attacks on Mobile Edge Computing Services," in IEEE Transactions on Mobile Computing, vol. 22, no. 1, pp. 1-18, 1 Jan. 2023, doi: 10.1109/TMC.2021.3086219.

[15] Z. Zhang, B. Guo, W. Sun, Y. Liu and Z. Yu, "Cross-FCL: Toward a Cross-Edge Federated Continual Learning Framework in Mobile Edge Computing Systems," in IEEE Transactions on Mobile Computing, vol. 23, no. 1, pp. 313-326, Jan. 2024, doi: 10.1109/TMC.2022.3223944.

[16] T. Ouyang, X. Chen, Z. Zhou, R. Li and X. Tang, "Adaptive User-Managed Service Placement for Mobile Edge Computing via Contextual Multi-Armed Bandit Learning," in IEEE Transactions on Mobile Computing, vol. 22, no. 3, pp. 1313-1326, 1 March 2023, doi: 10.1109/TMC.2021.3106746.

[17] Q. Li, X. Ma, A. Zhou, X. Luo, F. Yang and S. Wang, "User-Oriented Edge Node Grouping in Mobile Edge Computing," in IEEE Transactions on Mobile Computing, vol. 22, no. 6, pp. 3691-3705, 1 June 2023, doi: 10.1109/ TMC.2021.3139362.

[18] J. Liu, J. Ren, Y. Zhang, X. Peng, Y. Zhang and Y. Yang, "Efficient Dependent Task Offloading for Multiple Applications in MEC-Cloud System," in IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2147-2162, 1 April 2023, doi: 10.1109/TMC.2021.3119200.

[19] G. S. Park, R. H. Kim and H. Song, "Collaborative Virtual 3D Object Modeling for Mobile Augmented Reality Streaming Services Over 5G Networks," in IEEE Transactions on Mobile Computing, vol. 22, no. 7, pp. 3855-3869, 1 July 2023, doi: 10.1109/TMC.2022.3149543.

[20] J. He, S. Guo, M. Li and Y. Zhu, "AceFL: Federated Learning Accelerating in 6G-Enabled Mobile Edge Computing Networks," in IEEE Transactions on Network Science and Engineering, vol. 10, no. 3, pp. 1364-1375, 1 May-June 2023, doi: 10.1109/TNSE.2022.3190330.

[21] T. Kim, Y. Kim, E. Adu and I. Bang, "On Offloading Decision for Mobile Edge Computing Systems Considering Access Reservation Protocol," in IEEE Access, vol. 11, pp. 40457-40468, 2023, doi: 10.1109/ACCESS. 2023. 3269281.

[22] J. Lin, S. Huang, H. Zhang, X. Yang and P. Zhao, "A Deep-Reinforcement-Learning-Based Computation Offloading With Mobile Vehicles in Vehicular Edge Computing," in IEEE Internet of Things Journal, vol. 10, no. 17, pp. 15501-15514, 1 Sept.1, 2023, doi: 10.1109/JIOT.2023.3264281.

[23] X. Dong, Z. Di, L. Wang, Q. Yao, G. Li and Y. Shen, "Load Balancing of Double Queues and Utility-Workload Tradeoff in Heterogeneous Mobile Edge Computing," in IEEE Transactions on Wireless Communications, vol. 22, no. 7, pp. 4313-4326, July 2023, doi: 10.1109/TWC.2022.3224301.

[24] L. Zhang, X. Wu, F. Wang, A. Sun, L. Cui and J. Liu, "Edge-Based Video Stream Generation for Multi-Party Mobile Augmented Reality," in IEEE Transactions on Mobile Computing, vol. 23, no. 1, pp. 409-422, Jan. 2024, doi: 10.1109/TMC.2022.3232543.

[25] Min M, Liu Z, Duan J, Zhang P, Li S. Safe-Learning-Based Location-Privacy-Preserved Task Offloading in Mobile Edge Computing. *Electronics*. 2024; 13(1):89. https://doi.org/10.3390/electronics13010089