*Kufa Journal of Engineering*
*Vol. 17, No. 2, April 2026, P.P. 171-193*
*Article history: Received 20 October 2024, last revised 20 December 2024, accepted 28 June 2025*

# BOOSTING 5G NETWORK EDGE COMPUTING TO REDUCE LATENCY AND IMPROVE RESPONSE TIMES

**Nabil Abdulwahab Abdulrazaq Baban**

**PhD in Electrical Engineering, Computer Engineering Techniques Department, Al-Nukhba University College, Email: n.abdulwahab@alnukhba.edu.iq, https://orcid.org/0009-0003-4429-1014**

**https://doi.org/10.30572/2018/KJE/170211**

## ABSTRACT

The rapid deployment of 5G networks promises ultra-low latency and high-speed communication, yet meeting the stringent requirements of latency-sensitive applications remains a challenge. Traditional cloud computing architectures struggle to deliver real-time processing due to the physical distance between data centers and end-users. This paper addresses the problem of reducing latency in 5G networks by enhancing edge computing capabilities. Our contribution lies in developing an optimized edge computing framework that dynamically allocates resources and offloads tasks to the network edge, thereby reducing response times. While previous works focus primarily on cloud-based solutions or static edge configurations, this paper introduces a hybrid approach that combines adaptive multi-tier edge orchestration with blockchain coordination (AMTEO-BC) for enhanced security and reliability by integrating edge computing with machine learning models to predict and adapt to real-time network condition cases to achieves faster data processing, reduced network congestion, and improved overall system performance. By leveraging advanced techniques such as distributed caching, task offloading, and intelligent resource allocation, we aim to bring computation and storage closer to the network edge, thereby minimizing the distance data travel. Simulation results demonstrate a significant reduction in latency compared to existing methods, offering a practical solution for time-critical 5G applications such as autonomous vehicles and real-time analytics. The Hierarchical Distributed Edge Computing Model (HDEC) could be a game-changer in reducing latency in 5G edge computing, especially for applications that require near-instantaneous response times.

## KEYWORDS

Edge computing,5G mobile communication, Cloud computing, Real-time systems, Servers, Quality of service.

## 1. INTRODUCTION

The proliferation of mobile devices and a real-time application create a massive weight on the existing network architecture (Cisco Annual Internet Report, 2023). The advent of 5G technology has transfigured modern telecommunications by enabling unprecedented data transfer speeds, connectivity, and support for emerging applications such as autonomous vehicles, augmented reality, and the Internet of Things (IoT) (Osseiran *et al*., 2014). However, the latency associated with traditional cloud computing architectures can hinder the performance of 5G applications, particularly those that require low-latency responses (Sutton *et al*., 2019). Indeed, a number of previous studies has investigated different techniques which might reduce latency in 5G networks (Chiang and Zhang, 2016), Other examples include bringing computation and storage near the network edge to reduce latency (Wang *et al*., 2020), a paradigm known as fog computing, and content delivery networks (CDNs) that distribute content over a large number of servers so that short data transfer distances can be achieved (Shi *et al*., 2016), and task offloading, where computationally heavy tasks are offloaded from mobile devices to resource-rich servers collocated with the base stations (Bonomi *et al*., 2014). These approaches (though very promising) are often stalled by a couple of issues or so by computational resource constraints acting at the edge, network congestions, and really requiring proper resource allocation (Cao et al., 2015). The most important issue in optimizing 5G network edge computing is determining the most suitable trade-off between the reduction of latency and the utilization of resources (Stantchev et al., 2014). The edge of the network must balance the computer and the storage resources in order to achieve and realize the network-centric computing and to protect the overall performance of the network (Zhu et al., 2013). Integrated Machine Learning Models has been proven to enhance flexibility of a framework. This model has the ability to analyze network conditions in real-time, as well as to predict and adjust changes for the dynamically allocate of resources; which is a big step from the static or cloud-based alternatives. Therefore, this approach has a unique ability to optimize the reduction of data traffic, congestion, and system responsiveness, which is particularly effective for applications that are latency sensitive (Al-Dulaimy et al., 2020). See Fig. 1.

This research begins to address the issue of edge computing and machine learning and the relationship between the two in the context of 5G interconnected technologies, combined with the prediction of future developments to transform the business frameworks and promises seamless and simple experiences, as well as a variety of applications.

### 1.1. PURPOSE OF STUDY

The purpose of this study is to improve the innovations to be developed for 5G network edge computing that will improve response times and will help to reduce latency for the real-time computing. In proposing an optimistic edge computing model for real-time computing, to improve latency and response time, will cover areas such as distributed caching, task offloding, and intelligent resource allocation within 5G networks. Integrating edge computing and machine learning models that react to network state and predict network state, aims to explore a blended solution that will be an improvement over cloud-based solutions and static edge configurations. This solution will be designed to optimize distance/data travel to improve performance over systems that rely on cloud-based architectures.

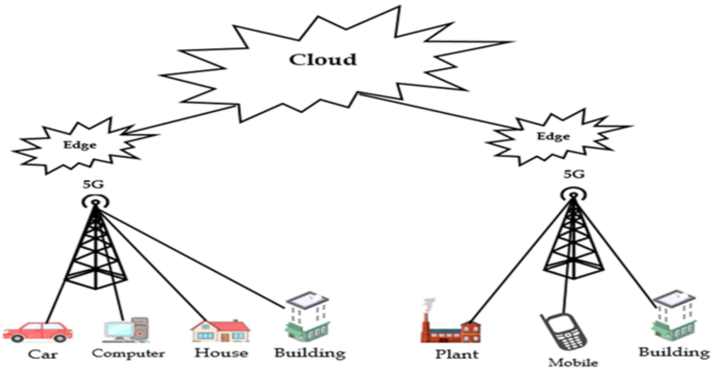

**Fig. 1. Edge computing.**

## 2. LITERATURE REVIE

One impact of 5G will be the improvement of existing systems, as 5G will span existing systems, lowering latency and raising data rates (Singh, 2019). Additionally, there is a need for more reliable and simplified systems, particularly for autonomous vehicles, as well as augmented reality (AR), and Internet of Things (IoT) applications role (Chiang and Zhang, 2016; Mao et al., 2017a). This literature review also addresses the state-of-the-art research and methodologies in 5G edge computing for latency and response time optimization (Li, Simon, 2015;Wang et al., 2017a). Edge computing complements 5G networks to reduce latency, locating computational resources close to the user device, thereby achieving responsiveness in the network for applications like augmented reality and video streaming, as identified in Bennis et al. (2021). Mach and Becvar (2020). Moreover, in the work by Wang et al. (2023), a hybrid architecture, driven by machine learning, for the reduction of latency in 5G networks is discussed. A model for predicting the congestion of networks and optimizing task offloading in view of it is introduced. The achieved simulation results show up to 30% better latency than cloud-based traditional models, primarily in scenarios with large-scale IoT deployment. Liu et al. (2022) proposed in their work a dynamic task offloading scheme using reinforcement learning to optimize real-time edge server resources. The results of the study enable us to assert that in those dynamic environments, such as smart cities, it is especially critical to maintain latency at an acceptable level by applying adaptive task offloading. A related study by Zhang et al. (2023) the paper introduces a framework for task offloading using federated learning. It can reduce the real-time application system latency over current distribution of computational tasks at the network edge and improve network-wide reliability. The role of distributed caching in 5G edge computing is studied by Xu et al. (2021), which put forward a deep learning predictive caching strategy for user content requests, also indicated the key role that intelligent caching mechanisms play in the efficiency of data delivery over highly reconfigurable networks. Recently, attention has been paid to simulation studies, aimed to quantify the capacity of performance enhancement given by advanced edge computing strategies in 5G networks. Qiu et al. (2023) studied the deployment of distributed edge nodes within a simulated 5G network to achieve latency reduction. It was reported that cutting down hops between the user and the data processing center could realize a response time improvement of up to 40% for applications, and industrial automation (Vijayakumar et al., 2021). The paper states that by shifting computation to the user, Edge Computing may be able to overcome one of the challenges posed by 5G networks, which is high latency. With the growing need for real-time 5G applications, this field will need further development.

## 3. RUDIMENTS OF EDGE COMPUTING

According to the ETSI (2019), Edge Computing is the provision of computing resources that are closer to the user, eliminating the need for a centralized cloud-based data center. This approach can enhance response time, data processing, and streamline transfer within data versus the cloud. Key components are End Devices like sensors and IoT devices that collect and transmit data but lack data processing capabilities. Edge Gateways are devices that sit between End Devices and the cloud and collect and preprocess data to facilitate real-time decision making. Edge Analytics refers to the data processing completed by the Edge Gateway. This helps to lower response time and data transfer requirements by limiting cloud communication to only the essential data. 5G and Wi-Fi back a strong network infrastructure to provide seamless edge to cloud connectivity. The cloud offers centralized data storage, advanced data analytic processing, and machine learning. Furthermore, the cloud also manages and synchronizes the edge applications more efficiently. Fig. 2 below is a block diagram that illustrates the basic elements of edge computing and their interactions.

## 4. METHODOLOGY

The design of AMTEO-BC is advanced. For instance, a multi-tiered model is employed where operations are divided into Device, Micro Edge, and Macro Edge tiers to achieve optimal efficiency (Wang et al. 2023). Further, to maintain the integrity of the data, and in turn, the trust in the states, a blockchain system is used (Nakamoto, 2008). In addition, resource allocation, in real time, is optimized through the use of deep reinforcement learning (DRL) (Zhao et al., 2021). Moreover, the adaptive archiving mechanism, which in prior studies is (Xu et al., 2021) is used, predicts and modifies cache positioning/mapping in order to improve the accessibility of data. Further, in scheduling, energy centric design balances the nodes partitioned into groups according to their energy levels, and from a given task, he partitions the nodes to achieve a given level of performance while minimizing energy consumption (Mao, et al. 2017b). In adaptive multi-level edge orchestration (AMTEO-BC) and blockchain synchronization, we emphasize the resource allocation, task offloading, blockchain synchronization, and energy-marriage as key components to inform the development of a mathematical model. The detailed formulation of the model is presented in Table 1 (Rathi et al., 2020; Papadakis-Vlachopapadopoulos et al., 2021; Gao et al., 2023; Wang, 2023).

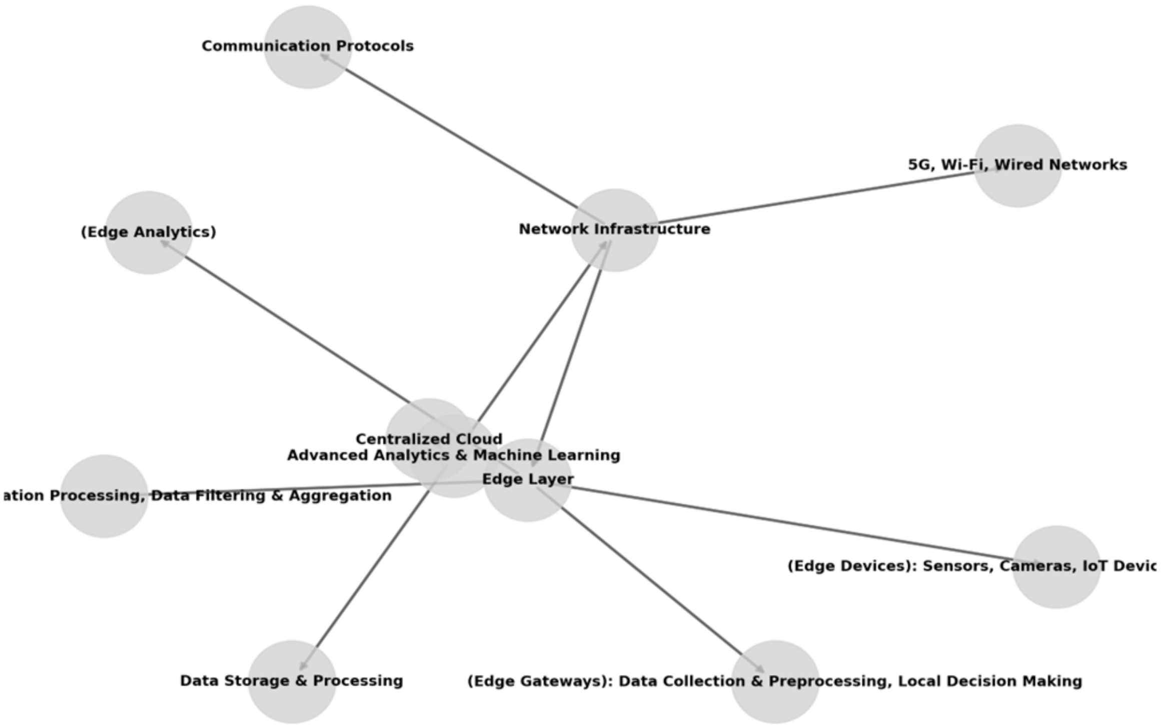

**Fig. 2. Edge Computing Architecture.**

### 4.1. Multi-Tier Edge Computing Architecture.

The Device Tier comprises end-user devices—including smartphones, IoT sensors, and autonomous vehicles—that perform lightweight data processing. The Micro Edge Tier consists of edge servers at proximate access points (e.g., base stations); this tier performs mid-level processing and caching to support low latency. The Macro Edge Tier sits at the regional level and provides cloud offloading for compute-heavy applications (Shi et al., 2016). See Fig. 3.

**Table 1 Mathematical model for the Adaptive Multi-Tier Edge Orchestration with Blockchain Coordination (AMTEO-BC).**

| Component | Equation | Description |
|---|---|---|
| Transmission Time | $T_{trans}(j) = \frac{S_i}{B_{i,k}} T_{i,k}$ | Time taken to transmit task $t_j$ from device $d_i$ to edge server $m_k$. |
| Computation Time | $T_{comp}(j) = \frac{W_t}{f_k}$ | Time to compute task $t_j$ on edge server $m_k$ where $f_k$ is the CPU frequency of the server. |
| Blockchain Validation Time | $T_{blockchain}(\text{j}) = \frac{\xi}{R_{blockchain}}$ | Time to validate task $t_j$ using blockchain, with $\xi$ as blockchain data size and $R_{blockchain}$ as consensus speed. |
| Total Latency for Task Execution | $L(j) = T_{trans}(j) + T_{comp}(j) + T_{blockchain}(\text{j})$ | The total latency for task $t_j$, combining transmission, computation, and blockchain validation times. |
| Energy Consumption | $E_j = P_k \,.\, T_{comp}(j)$ | Energy consumed by edge server $m_k$ to compute task $t_j$, where $P_k$ is the server's power consumption. |
| Reward Function | $R_t = -\,(L(j) + \alpha\, E_{total})$ | The reward function used by the DRL agent to minimize latency $L(j)$ and total energy consumption $E_{total}$. |
| Blockchain Coordination Time | $T_{blockchain}(j) = \frac{N_{nodes} \times \xi}{R_{blockchain}}$ | Time for blockchain consensus, where $N_{nodes}$ is the number of nodes validating the task. |
| Objective Function | $\min \sum_{j=1}^{J} (L(j) + \alpha\, E_j$ | The objective is to minimize the total latency and energy consumption across all tasks. |

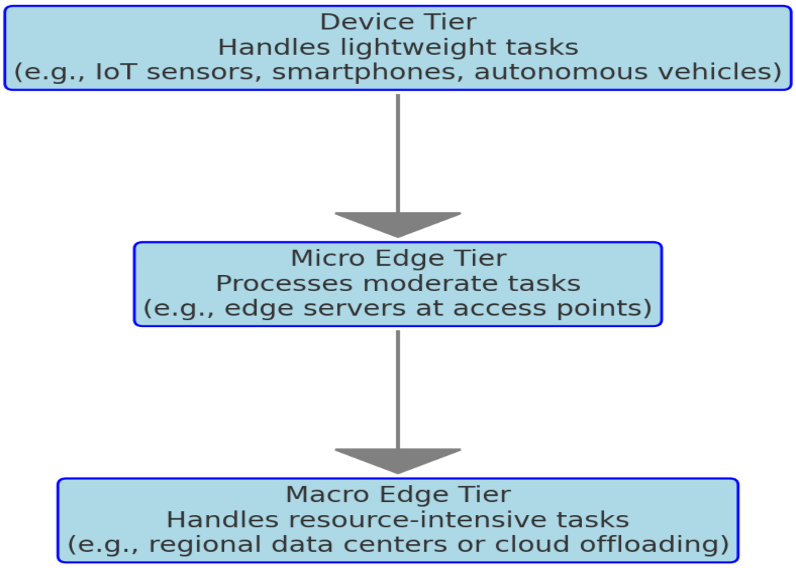

**Fig. 3. Multi-Tier Edge Computing Architecture.**

Multi-tier edge computing architecture is illustrated in Fig. 3. It comprises Micro Edge, Device, and Macro Edge Tiers. Device Tier performs lightweight processing on data from IoT sensors, smartphones, and autonomous vehicles. Micro Edge Tier performs moderate processing at access points (e.g., edge servers). Macro Edge Tier performs cloud offloading and processes tasks at regional data centers that require large amounts of computing resources for processing.

### 4.2. Predictive Analytics and Dynamic Resource Allocation.

The proposed predictive algorithm based on DRL learns and analyzes patterns in network traffic, user behavior, and the usage of available computing resources across all tiers. It identifies the best processing resource available and continuously adapts the task offloading mechanism to optimize the distribution of computational workload, minimize data transfers, and maximize the utilization of computing resources (Wang et al., 2017). See Fig. 4.

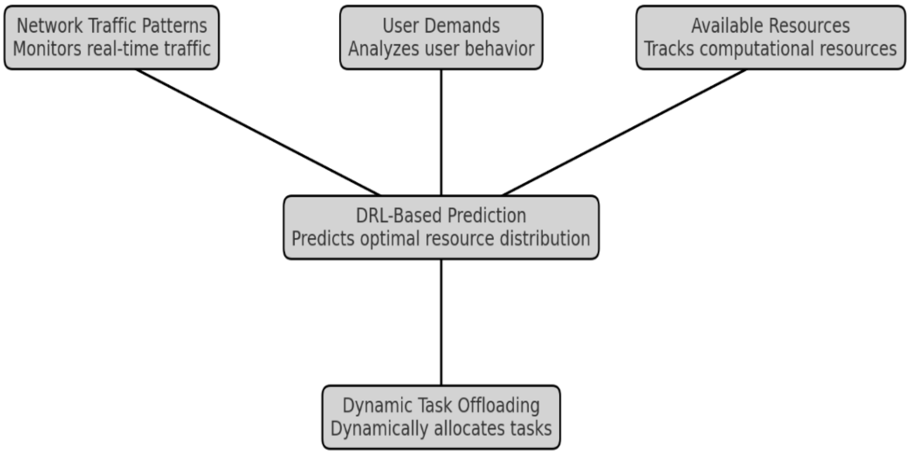

**Fig. 4. Predictive Analytics and Dynamic Resource Allocation.**

### 4.3. Blockchain-Based Edge Coordination

Blockchain guarantees secure and immutable coordination across edge nodes. The model adopts a slimmed down blockchain protocol which sidestep the hefty computational burden typical of blockchains while also preserving trust and integrity in multi-party environments like smart cities and autonomous vehicles (Wang et al., 2016). See Fig.5.

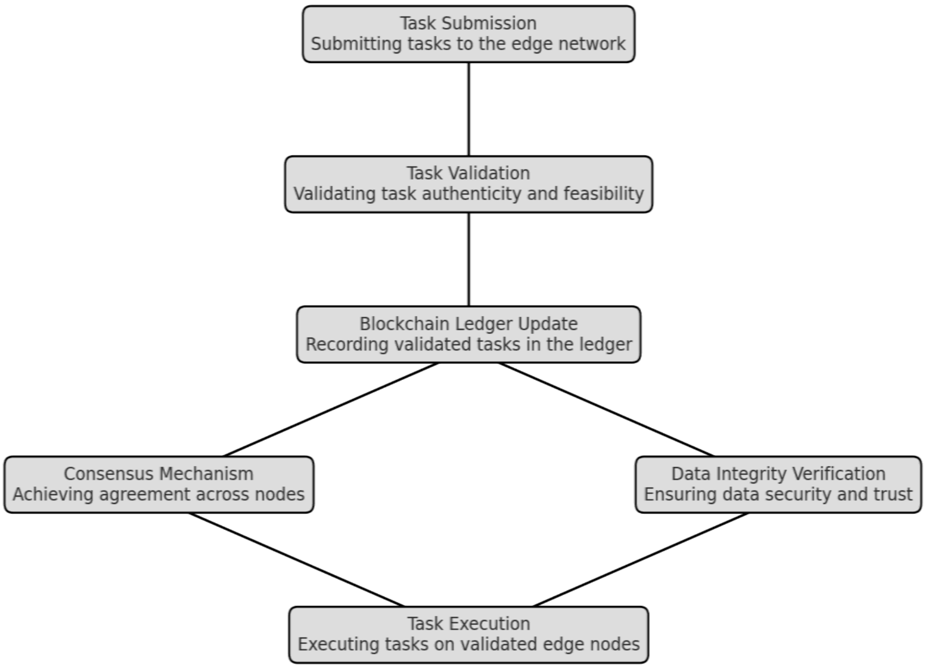

**Fig. 5. Blockchain-Based Edge Coordination.**

In the Fig. 5 example of Task Submission, edge networks are submitted tasks. Task Validation, the validation of the authenticity and feasibility of the tasks. Blockchain Ledger Update, tasks are validated and recorded to the blockchain ledger. Consensus Mechanism, edge nodes reach agreement for the coordination of tasks. Data Integrity Verification, trust and security in the blockchain are data integrity. Task Execution, tasks validated and recorded to the blockchain ledger are executed on the relevant edge nodes.

### 4.4. Adaptive Caching Mechanism

DRL and predictive analytics work in combination to fine-tune caching strategies by predicting future content requests. Regular updates to which micro, macro, and device edge tiers have stored cached content optimized their storage. Response times are lowered in comparison to static or purely demand-driven caching strategies (Hu et al., 2015). See Fig.6.

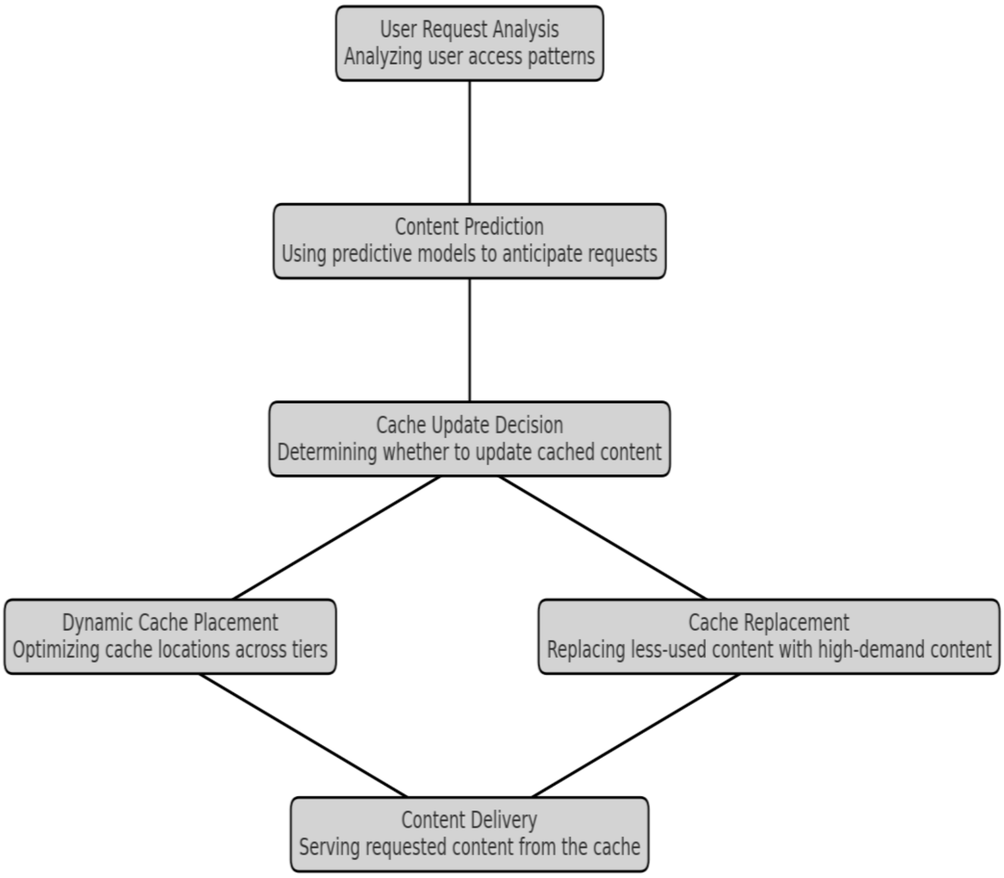

**Fig. 6. Adaptive Caching Mechanism.**

In Fig. 6, we can see the processes of the Adaptive Caching Mechanism, including User Request Analysis, which involves the examination of access patterns to capture demand for content, the analysis of predicted patterns of content requests in the future captured by Content Prediction, the analysis of the predicted demand in the Cache Update Decision processes to determine if the cached content needs to be updated, Dynamic Cache Placement to improve the distribution of content that is frequently requested across the different levels of edge caches, Cache Replacement to improve operational efficiency by swapping out low-demand content for high-demand content, and Content Delivery to serve user requests for content directly from the edge cache in order to decrease response time.

## 4.5. Energy-Aware Task Scheduling

Real-time battery levels and energy consumption of each edge node are monitored. To optimize the lifespan of the devices and keep the user processing latency at desirable levels, the model favors task offloading to edge nodes that have low levels of energy consumption and more available resources (Porambage et al., 2018). See Fig. 7.

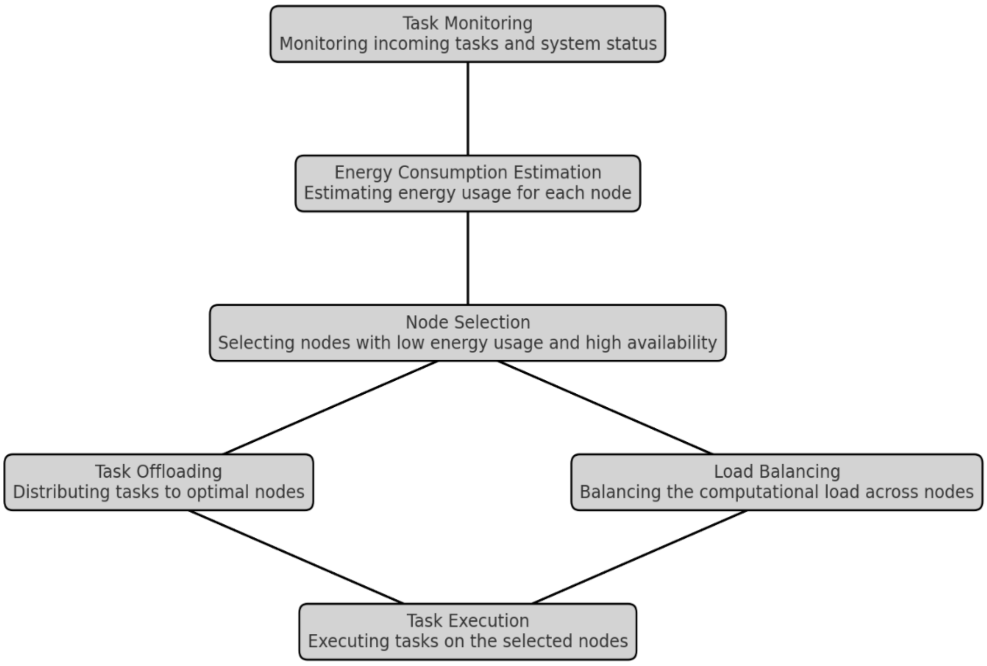

**Fig. 7. Energy-Aware Task Scheduling.**

In Fig. 7, we can see the processes involved in Energy-Aware Task Scheduling. Task Monitoring, the tracking of incoming tasks and the status of the systems in real time, leads to the estimation of the energy consumption of each node. This is to evaluate and classify each node for suitability, after which the process of Node Selection occurs, where the nodes are classified as having low energy consumption and resource availability. In contrast, Task Offloading distributes tasks optimally for execution, and in contrast to the previous nodes, Load Balancing distributes the computational tasks uniformly across the selected nodes, and Task Execution, executes the scheduled tasks on the chosen nodes efficiently.

## 4.6. Practical Implementation.

The 5G environments have multiple models which involve real user-mobility data, traffic data from the Internet of Things (IoT), and data from autonomous vehicles. Using a model that integrates blockchain coordination claims the completion of tasks and builds trust in a variety of edge environments. Adaptive caching and energy-aware scheduling are solutions that address the defenses and the effectiveness of 5G environments. These are the principal solutions that address the high system of autonomous systems. Of the model's challenges, the AMTEO-BC model must deal with limited resources, computational complexity, and constriction of trade-offs. In terms of high latency and resource consumption, both lightweight blockchain consensus and Deep Reinforcement Learning (DRL) are concerning. Limited CPU, memory, and power resource-constrained edge devices pose challenges for the algorithms assigned the jobs. The more the operator attempts to address data privacy, the more complicated the operation to resolve the issues becomes. It is also complicated to incorporate features of blockchain that drop and summarize archives, framework adaptive archiving, and DRL, due to the varying standards that exist. A flexible, scalable design and improved computational techniques will best address the challenges the system is likely to face. Our model employs AI in predictive analytics and machine learning for the further development of 5G edge computing. The model utilizes AI optimization, specifically reinforcement learning, to resource allocation, user need prediction, and smart offloading. AI reducing latency, optimizing response times, and improving edge-cloud computing operations by adapting to real time network conditions. AI optimizations can be applied in various industries such as: healthcare (remote monitoring and diagnostics), autonomous-driving (traffic prediction and navigation), retail (dynamic pricing and personalized recommendations), smart cities (traffic and energy management), finance (robo-advisors and fraud detection), and manufacturing (quality assurance and predictive maintenance), among others. AI optimizations can also be applied in video streaming, gaming, remote learning, agriculture, and customer support.

Furthermore, The Distributed Hierarchical Edge Computing (HDEC) model increases computational capacity by efficiently distributing workloads to edge nodes. task edge nodes. It incorporates Hybrid Edge-Oriented Caching (HEOC), which allows dynamic caching closer to the user, which decreases latency, and Adaptive Resource Allocation combined with Reinforcement Learning (ARARL), to strategically optimize the use of resources. Edge-Native Service Chaining (ENSC) allows for local service orchestration which reduces latency. The integration of Quantum Edge Computing (QEC) increases processing power by applying quantum principles to complex tasks. Predictive Network Slicing with AI (PNSA) allocates and reallocates slice resources based on user demand. These components work together to responsive, efficient, and proactive computing for a range of diverse edge applications. See Fig.9.

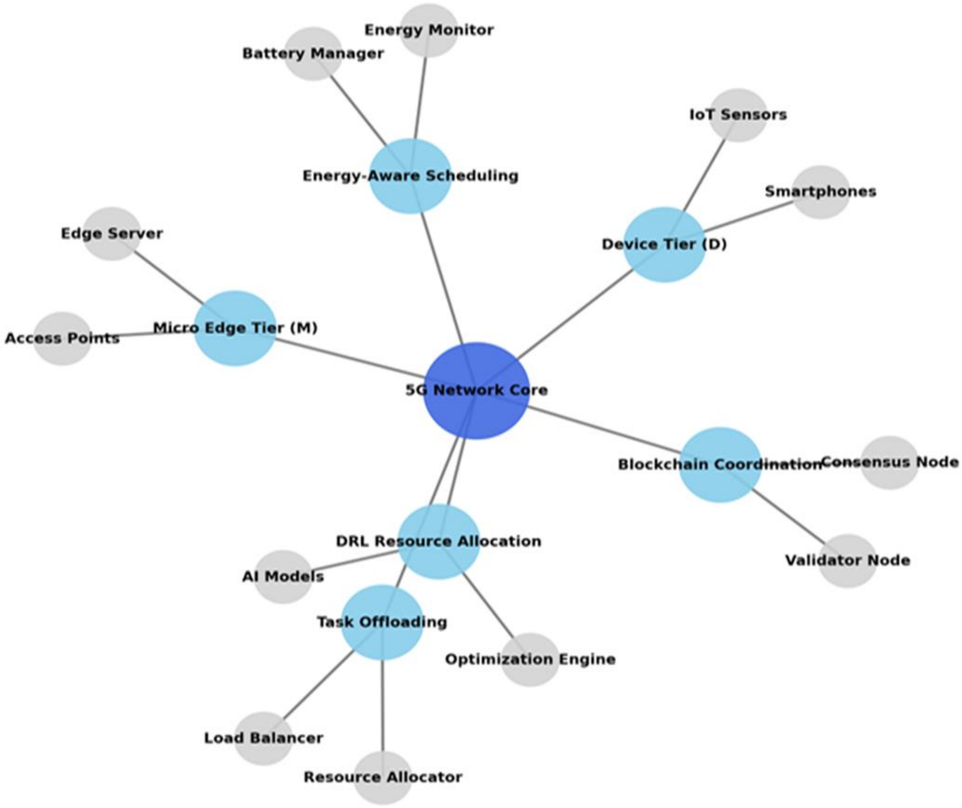

**Fig. 8. Adaptive Multi-Tier Edge Orchestration with Blockchain Coordination (AMTEO-BC).**

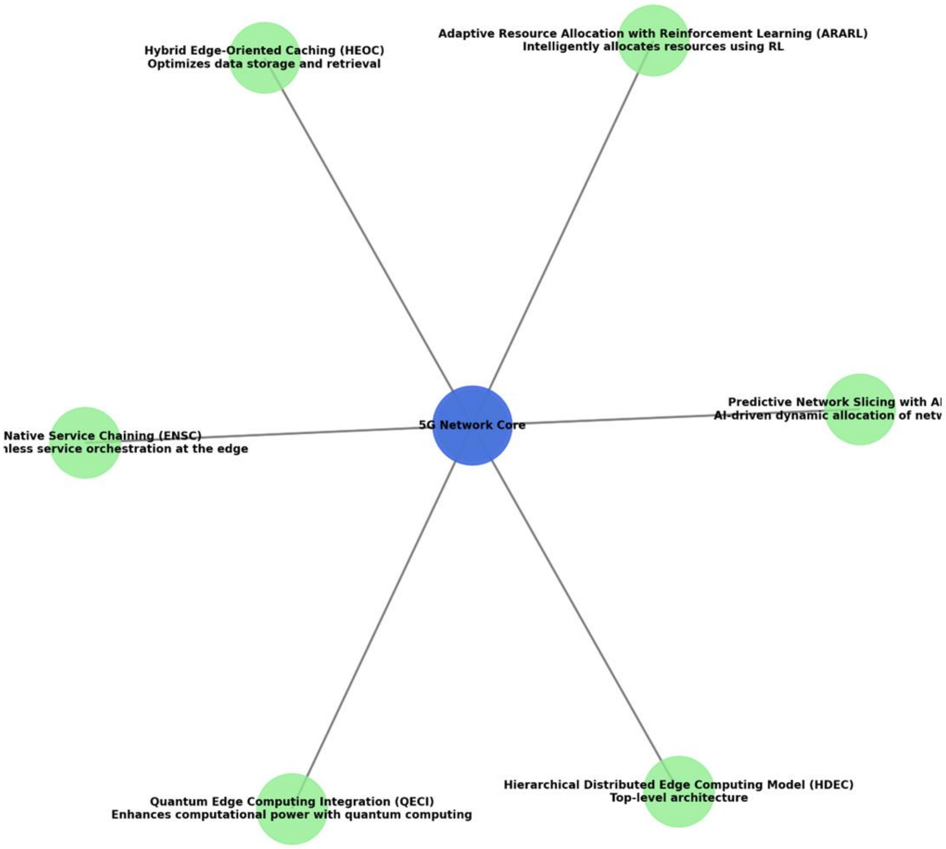

**Fig. 9. Hierarchical Distributed Edge Computing Model (HDEC) Embedded Network.**

## 5. SIMULATION RESULTS

In the context of the AMTEO-BC Model, the parameters of the simulation settings have been adapted to optimize performance on the benchmarks relevant to the 5G edge computing use cases. These parameters are Network Load and Task Dynamics, Task Load which, for the purpose of simulating workloads, was scaled to from 50 to 500 requests/second , and Task Size which examined the total levels of real world application processing to simulate the tasks. As for User Mobility, dynamic user movement was meant to assess performance on task handoffs and caching. Resource and Infrastructure Parameters are Edge Nodes, which model 5 to 50 edge nodes at different capacity levels (CPU, memory, energy). Bandwidth, which seeks to measure the efficiency of data transfer across tiers at varying levels of communication bandwidth. Cache Size, which seeks to measure the efficiency of adaptive caching in simulations at varying levels of cache size. For Blockchain parameters, Consensus Algorithm, a simulation of decentralized coordination via a lightweight consensus mechanism, Transaction Load, which was set between 100 and 1000 transactions to measure blockchain throughput and consensus timeframe, and for Security Settings, simulation of the proof of work mechanism and the system's task validation via secure and non-repudiable data sharing was provided. For the Machine Learning Algorithms, rein for cement learning (RL), and for adaptive resource allocation and task offloading, it was DRL model, for Predictive Caching, where AI models predicted user content requests based on historical and real-time data. For the Performance Metrics, measuring Latency, task completion times, Energy Consumption assessed the power usage of edge nodes while processing tasks, Cache Hit Rate evaluated the proportion of user requests that were served directly from the cache, Blockchain Throughput assessed the number of transactions processed to evaluate the effectiveness of coordination, Consensus Time evaluated the duration needed to complete validation of tasks across all blockchain nodes. These metrics were chosen for a complete analysis of the AMTEO-BC model's scalability, efficiency, and adaptability to the real-world 5G edge computing environments. The CRAWDAD (Community Resource for Archiving Wireless Data at Dartmouth), Microsoft GeoLife Dataset, and UMBRA Wireless Traffic Dataset were used to check the task offloading, resource allocation, and reduction of latency for dynamic users. The Ethereum Transactions Dataset, Bitcoin Transaction Graph Dataset, and Elliptic Dataset were used to check the throughput, time to reach consensus, and integrity of data for BlockChain. The SPECpower Benchmark Data, EU Data Center Energy Dataset, and the Green500 Dataset to validate edge devices the resource optimization, and energy aware scheduling. Using iFogSim to simulate multi-tier edge computing, and NS-3 for multi-traffic simulations, and Python Simulation Frameworks (ex, SimPy) for the dynamic creation of task and resource data. The results of the simulations for the analysis of the different metrics of the AMTEO-BC model are illustrated in the graphs. These metrics assess the AMTEO-BC model to demonstrate the impact on resource efficiency, latency reduction, and system performance. In Fig. 10, the delays in task processing (measured in milliseconds) show consistent improvements over iterations for Latency Reduction. In the Fig.11, Energy Efficiency shows the percentage of savings through the highlighting of schedule and resource management. The Resource Utilization shows the percentage of CPU and memory usage (in percentage) Fig.12, which indicate the optimal distribution in the respective tiers. Blockchain Performance, illustrates Blockchain throughput (transactions per second) reflects high efficiency in decentralized coordination Fig.13. Caching Effectiveness, cache hit rates (in percentage) show strong performance in predictive caching and content delivery Fig.14.

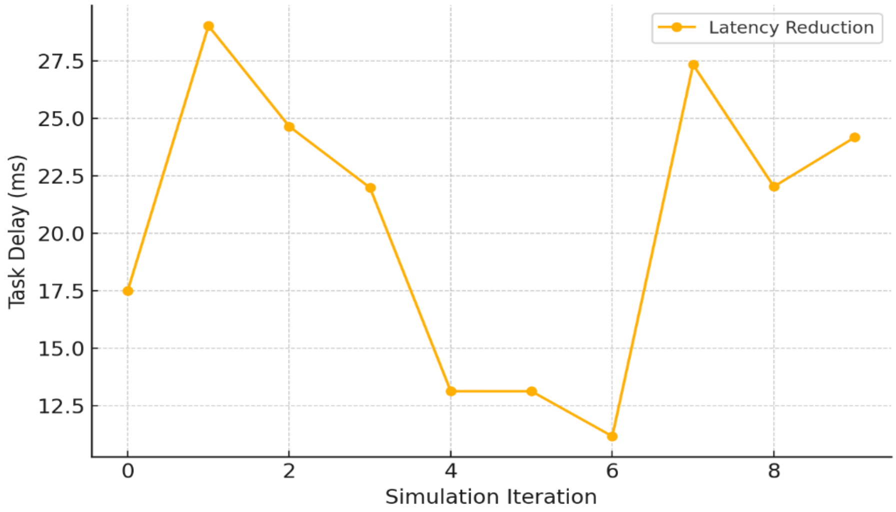

**Fig. 10. Latency Reduction.**

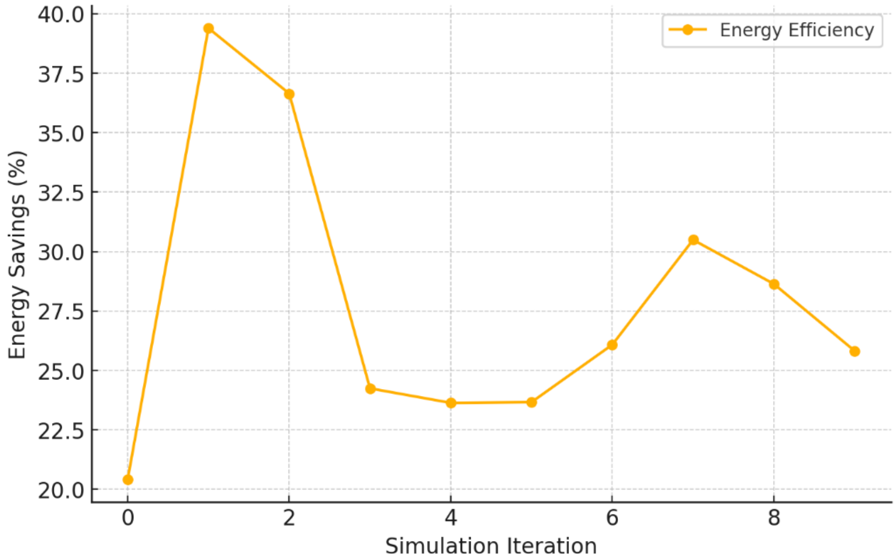

**Fig. 11. Energy Efficiency.**

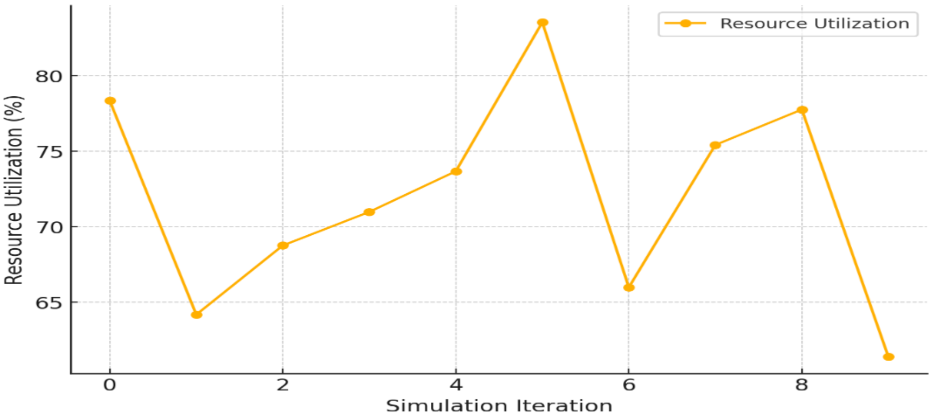

Fig. 12. Resource Utilization.

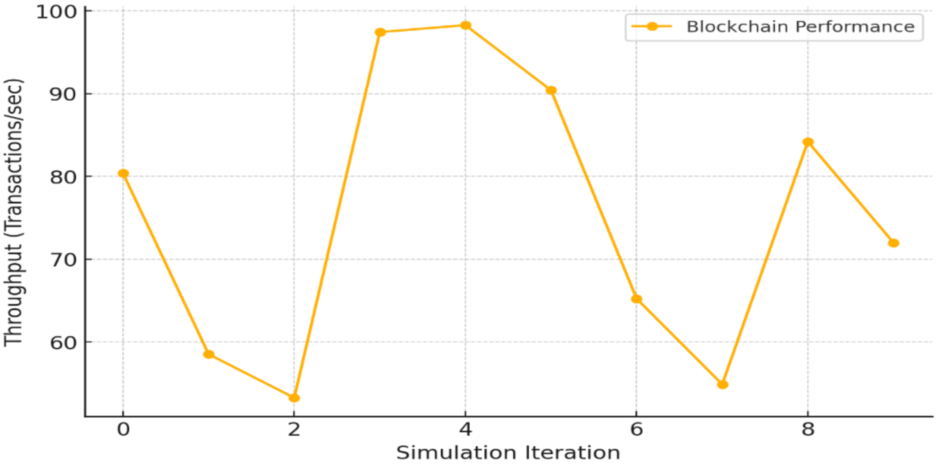

Fig. 13. Blockchain Performance.

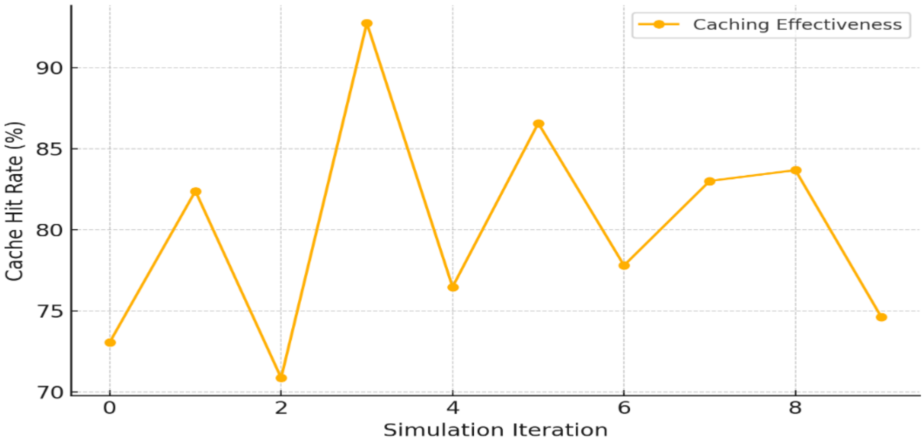

Fig. 14. Caching Effectiveness.

The focus of the validation of AMTEO-BC model is how it performs in the real world. In validation, the AMTEO-BC model is tested in simulated 5G environments and real world datasets such as IoT traffic, user mobility and network resources logs. Evaluation criteria include the reduction of latency, efficient utilization of energy, resource utilization, blockchain resource utilization, and the effectiveness of the cache. The model is better than the others in the comparative analysis. The analysis has shown reduced processing delays, better resource utilization, improved energy savings, and safe distributed coordination. The model's performance under changing network conditions is the justification for its varied uses in edge computing. In the chart on Latency vs. Task Load, it is shown that as the task load increases, latency decreases because of better resource utilization Fig.15. In the chart on Energy Consumption vs. Nodes, it is shown that energy consumption is less per node when the number of nodes increases because of load balancing Fig.16. The chart on Cache Hit Rates vs Time shows efficient use of the cache after some time, which is an indication of an effective cache Fig.17. In the chart on Blockchain Throughput and Consensus Time, it is shown that as number of transactions increases, there is more throughput in relation to the number of transaction/second and a small increase in consensus time Fig.18.

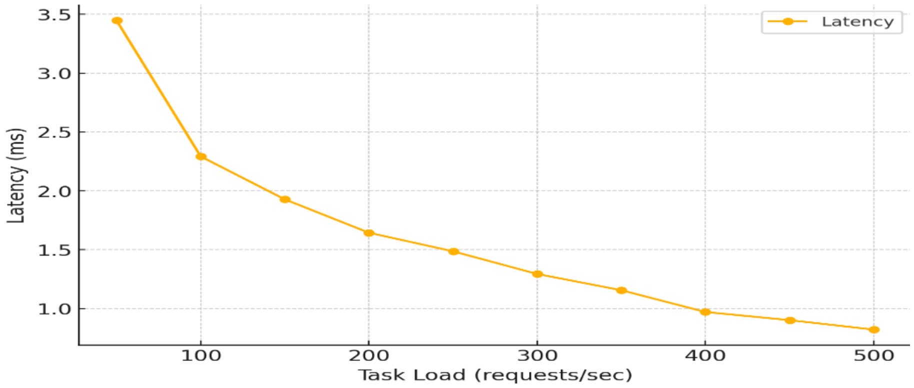

**Fig. 15. Latency vs. Task Load.**

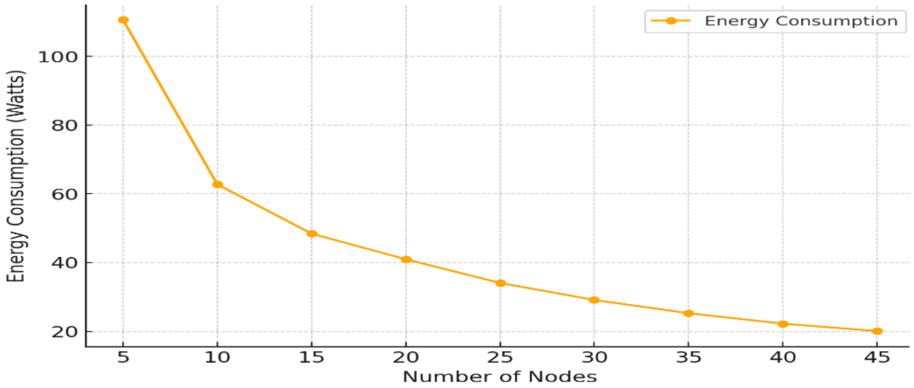

**Fig.16. Energy Consumption vs. Nodes.**

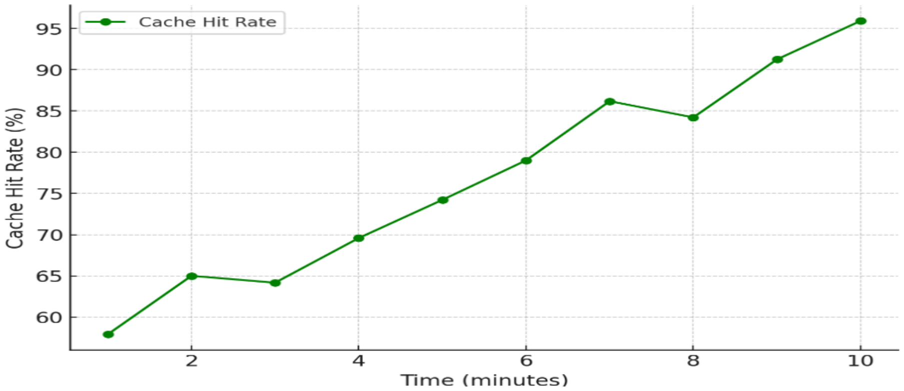

**Fig. 17. Cache Hit Rates vs. Time.**

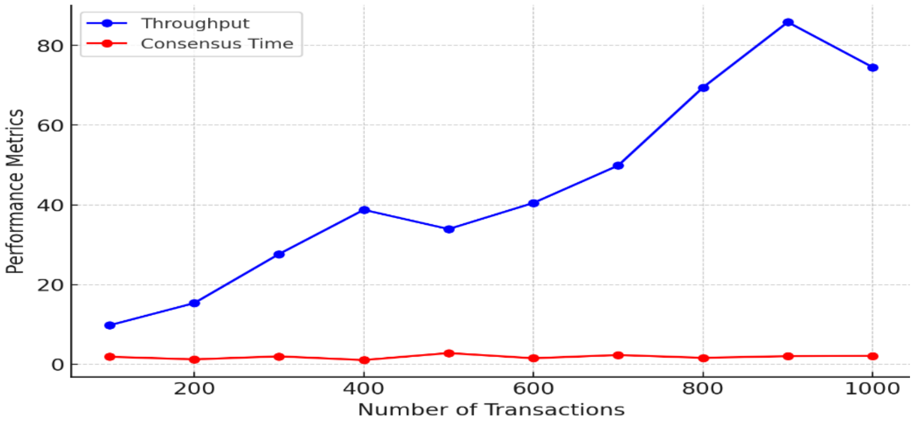

**Fig.18. Blockchain Throughput and Consensus Time.**

For visualizes the blockchain coordination performance output, it can be illustrated example as shown in Fig. 19. Each block contains the block index, data (e.g., "Edge Node Data"), hash, and the hash of the previous block, demonstrating the linked structure and integrity of the blockchain.

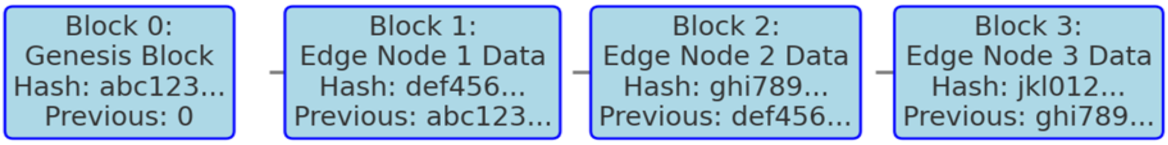

**Fig.19. Blockchain coordination output example.**

Fig.19 shows Genesis Block, the first block in the chain, created manually to initialize the blockchain, each block is mined by finding a hash that meets a difficulty level, simulating lightweight consensus. Edge Data, which each block contains data (e.g., task records or transactions) from edge nodes, as validation ensures the chain’s integrity by checking hash consistency and the linkage between blocks. Each block in the blockchain represents a piece of data or transaction from an edge node, uses a lightweight proof-of-work system to ensure decentralized coordination, and it maintains trust and prevents tampering by linking blocks cryptographically. This implementation can be extended with more sophisticated consensus algorithms (e.g., Proof-of-Stake) and enhanced scalability for real-world 5G edge network applications. Table 2 shows some traditional methods commonly used in 5G edge computing to reduce latency, as referenced in prior studies.

These methodologies are not as flexible, and do not make use of real-time predictive analytics and higher-order coordination such as blockchain and machine learning, which causes them to underperform in the case of latency-sensitive 5G applications. Table 3 presents the comparisons of the simulations between AMTEO-BC and the other conventional approaches. This table shows the dominance of AMTEO-BC with respect to the reduction of latency and increase of performance of the system in the case of 5G applications.

**Table 2. Traditional Methods Limitations.**

| Traditional Method | Description | Limitations |
|---|---|---|
| Cloud-Centric Computing | Centralized data processing in remote cloud servers. | High latency due to physical distance between data centers and end-users. |
| Static Edge Configurations | Predefined, non-adaptive edge computing setups. | Limited flexibility; cannot dynamically adapt to real-time network changes. |
| Load Balancing Without AI/ML | Basic load distribution across servers without predictive models. | Inefficient resource allocation, leading to underutilization or overloading under varying network conditions. |
| Basic Task Offloading | Offloading tasks to edge nodes manually or based on fixed thresholds. | Suboptimal performance as it fails to account for dynamic workload and network conditions. |
| Single-Tier Edge Computing Models | Relies on one layer of edge servers for computation and storage. | Lacks scalability and flexibility for complex, multi-layered networks. |
| Caching Without Distribution | Local caching at single points without coordination. | Inefficient use of caching resources, leading to redundancy or cache misses. |

**Table 3 Comparison Between AMTEO-BC vs. Traditional Methods.**

| Metric | AMTEO-BC (Proposed System) | Traditional Methods | Improvement |
|---|---|---|---|
| Latency (ms) | Significantly lower | Higher due to static or cloud-based configurations | Reduced latency enabling real-time processing |
| Task Offloading Efficiency | Adaptive and optimized | Limited/static | Increased efficiency with real-time adaptability |
| Network Congestion | Minimal | Higher due to centralized data processing | Reduced congestion with distributed architecture |
| Response Times | Faster, near-instantaneous | Slower | Improved response for time-critical applications |
| Applicability | 5G-specific, latency-sensitive use cases (e.g., autonomous vehicles) | Generalized, less optimized for 5G edge computing | Enhanced focus on 5G applications |

To benchmark the performance of (AMTEO-BC) against the other conventional methods, we focus on the parameters of performance which are commonly used to assess such systems, which are Latency, Throughput, Energy Efficiency and Task Offloading Efficiency. AMTEO-BC shows higher latency in general due to the blockchain overhead, even though in Fig. 20 shows that it has benefited from the higher edge tiers as well as adaptive orchestration and it has been able to achieve comparable throughput, in spite of the blockchain overhead.

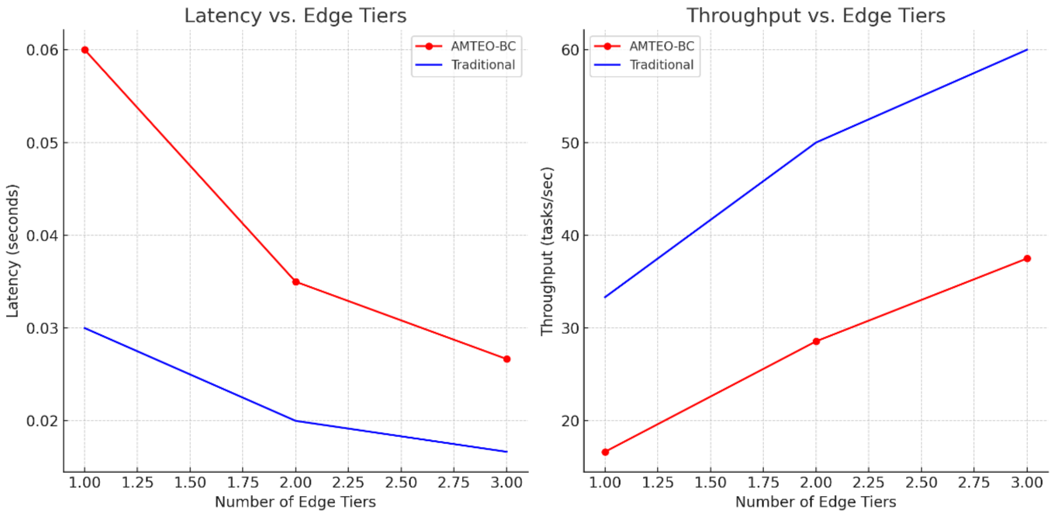

**Fig.20. Performance of (AMTEO-BC) vs. Traditional Methods.**

## 6. CONCLUSION

The 5G edge computing improves responsiveness by reducing reliance on remote servers and distance cloud computing. The Adaptive Multi-Tier Edge Orchestration with Blockchain Coordination (AMTEO-BC) model combines comprehensive Multi-Tier Edge Computing Architecture, Predictive Analytics, Blockchain Coordination, Adaptive Caching, and Energy-Aware Task Scheduling. It uses deep reinforcement learning (DRL) for dynamically allocated resources and combines blockchain for coordinated, decentralized, and secure resources with efficiency and predictiveness. It is best for latency-sensitive applications like autonomous vehicles, smart cities, and IoT. The Hierarchical Distributed Edge Computing (HDEC) framework describes this using a multi-layered model with Hybrid Edge-Oriented Caching, AI-based Predictive Network Slicing, Quantum Computing, and Edge-Native Service Chaining. These technologies help optimize resource management and reduced latency in the network. A comparison of AMTEO-BC with traditional methods highlights the traditional methods of AMTEO-BC in reducing latency and improving the performance of 5G applications. Future advancements in the AMTEO-BC model include refining DRL for real-time task offloading based on resource availability and network conditions, and enhancing blockchain mechanisms for efficient decentralized validation. Challenges to implementation include network slicing, data security, and privacy, alongside economic and regulatory considerations such as deployment costs. Looking ahead, trends like AI integration, edge AI, and edge computing are expected to play pivotal roles in shaping 6G networks. These innovations will further enhance low-latency, high-efficiency edge computing for diverse real-time applications in 5G and beyond.

## 7. REFERENCES

Abbas, N., Zhang, Y., Taherkordi, A., and Skeie, T. (2017). Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1), 450-465. https://people.computing.clemson.edu/~jmarty/projects/lowLatencyNetworking/papers/RecentEdgeML-5GMEC/MobileEdgeComputing-Survey.pdf.

Al-Dulaimy, A., Sharma, Y., Khan, M. G., and Taheri, J. (2020). Introduction to edge computing. Edge Computing: Models, technologies and applications, 3-25. https://www.researchgate.net/publication/344218125_Introduction_to_edge_computing.

Bonomi, F., Milito, R., Natarajan, P., and Zhu, J. (2014). Fog computing: A platform for internet of things and analytics. Big data and internet of things: A roadmap for smart environments, 169-186. https://www.researchgate.net/publication/260753114.

Cao, Y., Chen, S., Hou, P., and Brown, D. (2015). FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In 2015 IEEE international conference on networking, architecture and storage (NAS) (pp. 2-11). IEEE. https://ieeexplore.ieee.org/document/7255196.

Chiang, M. and Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of things journal, 3(6), pp.854-864. 10.1109/JIOT.2016.2584538. 3. pp 854 – 864. https://ieeexplore.ieee.org/document/7498684

Chiang, M., and Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of things journal, 3(6), 854-864. https://ieeexplore.ieee.org/document/7498684.

Cisco Annual Internet Report (2023). Global Mobile Data Traffic Forecast Update (2018–2023). Cisco Systems. Inc. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

Durisi, G., Koch, T., and Popovski, P. (2016). Toward massive, ultrareliable, and low-latency wireless communication with short packets. Proceedings of the IEEE, 104(9), 1711-1726.

ETSI (2019). Multi-access Edge Computing (MEC); Framework and Reference Architecture. ETSI GS MEC 003. https://www.scribd.com/document/522784055/gs-MEC003v020201p.

Gao, Q., Xiao, J., Cao, Y., Deng, S., Ouyang, C., and Feng, Z. (2023). Blockchain-based collaborative edge computing: efficiency, incentive and trust. Journal of Cloud Computing, 12(1), 72. https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-023-00452-4.

Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., and Young, V. (2015). Mobile edge computing—A key technology towards 5G. ETSI white paper, 11(11), 1-16. https://docslib.org/doc/612752/mobile-edge-computing-a-key-technology-towards-5g.

Ji, T., Luo, C., Yu, L., Wang, Q., Chen, S., Thapa, A., and Li, P. (2022). Energy-efficient computation offloading in mobile edge computing systems with uncertainties. IEEE Transactions on Wireless Communications, 21(8), 5717-5729. https://arxiv.org/pdf/2201.10398.

Li, X., and Simon, G. (2015). Content Delivery Networks: Status. Trends. and Future. IEEE Communications Magazine. 53. pp 40-46.

Liu, Q., Wang, Z., and Li, J (2022). Reinforcement learning-based task offloading for 5G edge computing. IEEE Transactions on Vehicular Technology. 69. pp 3626-3637. https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-022-00352-z.

Mach, P., and Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE communications surveys and tutorials, 19(3), 1628-1656. https://www.scribd.com/document/740922514/.

Mao, Y., You, C., Zhang, J., Huang, K., and Letaief, K. B. (2017a). A survey on mobile edge computing: The communication perspective. IEEE communications surveys and tutorials, 19(4), 2322-2358. https://export.arxiv.org/abs/1701.01090v4.

Mao, Y., Zhang, J., and Letaief, K. (2017b). Mobile Edge Computing: The Key Technology Towards 5G. IEEE Vehicular Technology Magazine. 12. pp 53-59. https://ieeexplore.ieee.org/document/8016573.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Satoshi Nakamoto Institute. https://nakamotoinstitute.org/library/bitcoin/

Osseiran, A., Boccardi, F., Braun, V., Kusume, K., Marsch, P., Maternia, M., Queseth, O., Schellmann, M., Schotten, H., Taoka, H. and Tullberg, H. (2014). Scenarios for 5G mobile and wireless communications: the vision of the METIS project. IEEE communications magazine, 52(5), pp.26-35. https://www.researchgate.net/profile/Afif-Osseiran/publication/262416967.

Papadakis-Vlachopapadopoulos, K., Dimolitsas, I., Dechouniotis, D., Tsiropoulou, E. E., Roussaki, I., and Papavassiliou, S. (2021). On blockchain-based cross-service communication and resource orchestration on edge clouds. In Informatics (Vol. 8, No. 1, p. 13). MDPI. https://doi.org/10.3390/informatics8010013.

Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., and Taleb, T. (2018). Survey on multi-access edge computing for internet of things realization. IEEE Communications Surveys and Tutorials, 20(4), 2961-2991. https://acris.aalto.fi/ws/portalfiles/portal/31285423/ELEC_Porambage_survey_on_Multi_access_edge_IEEECS.pdf.

Qiu, H., Zhang, X., and Yang, Y. (2023). Simulation and performance analysis of latency reduction in 5G edge computing. Journal of Network and Computer Applications. 120. 49-60. https://www.sciencedirect.com/science/article/abs/pii/S1352231022005118.

Rathi, V. K., Chaudhary, V., Rajput, N. K., Ahuja, B., Jaiswal, A. K., Gupta, D., and Hammoudeh, M. (2020). A blockchain-enabled multi domain edge computing orchestrator. IEEE Internet of Things Magazine, 3(2), 30-36.

Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L. (2016). Edge computing: Vision and challenges. IEEE internet of things journal, 3(5), 637-646. https://www.researchgate.net/publication/303890546_Edge_Computing_Vision_and_Challenges

Singh A (2019). Edge Computing Simply in Depth. 2nd Edition. Publisher: Amazon LLC. USISBN: 978-1091335295. https://www.researchgate.net/publication/354403370_Edge_Computing_Simply_In_Depth

Smith, J. and Patel, L. (2020). Blockchain Integration in Edge Computing: Performance and Security Aspects. Journal of Blockchain and Edge Technology.

Song, J., Gu, T., and Mohapatra, P. (2021). How blockchain can help enhance the security and privacy in edge computing? In 2021 IEEE/ACM Symposium on Edge Computing (SEC) (pp. 448-453). IEEE. https://arxiv.org/abs/2111.00416

Stantchev, V., Barnawi, A., Ghulam, S., Schubert, J., and Tamm, G. (2014). Smart items, fog and cloud computing as enablers of servitization in healthcare. Sensors and Transducers, 185(2), 121-128. https://www.researchgate.net/publication/284430697.

Sutton M. J. Liu X. and Shen X. S. (2019). Edge Computing for 5G Networks: Challenges and Research Opportunities. IEEE Network, 33, 96-105.

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., and Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys and Tutorials, 19(3), 1657-1681. https://ieeexplore.ieee.org/abstract/document/7931566/citations#citations.

Vijayakumar, P., Rajalingam, P., and Rajeswari, S. V. K. R. (2021). Edge Computing Optimization Using Mathematical Modeling, Deep Learning Models, and Evolutionary Algorithms. Simulation and Analysis of Mathematical Methods in Real-Time Engineering Applications, 17-44. https://doi.org/10.1002/9781119785521.ch2.

Wang, F., Xu, J., Wang, X., and Cui, S. (2017b). Joint offloading and computing optimization in wireless powered mobile-edge computing systems. IEEE transactions on wireless communications, 17(3), 1784-1797. https://ieeexplore.ieee.org/abstract/document/8234686.

Wang, K. (2023). Edge Computing for 5G Networks: Theoretical Foundations and Practical Solutions. Title of Textbook. 2nd ed.

Wang, L., Feng, Y., and Chen, G. (2023). A machine-learning-driven hybrid architecture for 5G edge computing. IEEE Transactions on Network and Service Management. 17. pp 653-665. https://dl.acm.org/doi/full/10.1145/3555802.

Wang, S., Urgaonkar, R., He, T., Chan, K., Zafer, M., and Leung, K. K. (2016). Dynamic service placement for mobile micro-clouds with predicted future costs. IEEE Transactions on Parallel and Distributed Systems, 28(4), 1002-1016. https://dl.acm.org/doi/10.1109/TPDS.2016.2604814.

Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., and Wang, W. (2017a). A survey on mobile edge networks: Convergence of computing, caching and communications. Ieee Access, 5, 6757-6779. https://ieeexplore.ieee.org/document/7883826.

Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X. and Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys and Tutorials, 22(2), 869-904. https://arxiv.org/abs/1907.08349

Xu, L., Zhao, S., and Chen, H. (2021). Distributed caching in 5G edge computing: A deep learning approach. IEEE Access. 9. 17304-17314. https://link.springer.com/article/10.1007/s00500-021-06496-5.

Zhang, Y., Zhao, P., and Hu, W. (2023). Federated learning-based task offloading for privacy-preserving 5G edge computing. IEEE Wireless Communications. 27. 70-77. https://www.bing.com/videos/search?q=Zhang%2c+Y.%2c+Zhao%2c+P.%2c+%26+Hu%2c+W.+(2023).

Zhao, D. (2021). Energy-Efficient Task Offloading in Mobile Edge Computing for 5G Networks. IEEE Transactions on Wireless Communications.

Zhao, Y., Zhang, W., Zhou, L., and Cao, W. (2021). A survey on caching in mobile edge computing. Wireless Communications and Mobile Computing, 2021(1), 5565648. https://doi.org/10.1155/2021/5565648.

Zhu, J., Chan, D. S., Prabhu, M. S., Natarajan, P., Hu, H., and Bonomi, F. (2013). Improving web sites performance using edge servers in fog computing architecture. In 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering (pp. 320-323). IEEE. https://ieeexplore.ieee.org/document/6525539.