Edge Computing in Mobile Networks: Enhancing Performance and Addressing Challenges

Anwer Jabbar Hasan

University of Misan

DOI:
https://doi.org/10.47134/jtsi.v2i3.4795
*Correspondence: Anwer Jabbar Hasan
Email: anwer.jabbar@uomisan.edu.iq

Received: 30-05-2025
Accepted: 30-06-2025
Published: 30-07-2025

Abstract: This research proposes a unified AI-based framework to enhance mobile network performance using edge computing. It introduces ARMA for latency reduction and CETO for energy optimization. Both algorithms rely on predictive analytics and adaptive task management. Implemented in Python and validated using NS-3 simulations and real telecom data, ARMA reduced latency by up to 50%, while CETO decreased energy use by 35%. Results were statistically significant ($p < 0.05$) across urban and rural scenarios. The framework provides a scalable, efficient, and secure solution for edge deployment, supporting real-time applications such as IoT and autonomous systems.

Keywords: Edge Computing, Latency Optimization, Energy Efficiency, AI-Based Task Management, Mobile Network Security

Introduction

The increasing demand for real-time services such as Internet of Things (IoT), augmented reality (AR), and autonomous systems has placed unprecedented stress on mobile network infrastructures. Traditional cloud computing architectures, with their centralized processing models, are unable to meet the low-latency and high-throughput requirements of these applications. This limitation has led to growing interest in edge computing, where data processing occurs closer to end-users.

Edge computing reduces transmission delays, improves bandwidth utilization, and supports faster response times by offloading tasks to nearby edge nodes. However, deploying edge computing in large-scale mobile environments introduces challenges related to resource allocation, energy efficiency, scalability, and data security. These issues are further complicated by the dynamic nature of mobile traffic and varying environmental conditions across urban and rural settings.

Previous research has addressed latency optimization and energy savings independently, but few frameworks offer an integrated solution that balances both. In addition, many existing models are validated only in simulated environments, lacking real-world deployment data. Security concerns, particularly those related to data integrity and unauthorized access, remain inadequately addressed in most edge computing studies.

This paper introduces an AI-based framework that simultaneously tackles latency reduction and energy efficiency while incorporating core security considerations. The framework consists of two key components:

- **ARMA** (Adaptive Resource Management Algorithm): predicts mobile traffic patterns and reallocates resources to minimize latency.
- **CETO** (Cost-Efficient Task Offloading): optimizes energy usage by dynamically distributing tasks between edge and cloud nodes.

Both algorithms are evaluated using NS-3 simulations and validated with real-world mobile network datasets. The study contributes a scalable and adaptable edge computing framework that addresses the core challenges of modern mobile networks and enables practical deployment across heterogeneous environments

Problem Statement

Despite the advantages of edge computing in reducing latency and enhancing network responsiveness, its deployment in real-world mobile environments remains limited. Most existing research focuses on optimizing a single performance aspect either latency or energy consumption without offering an integrated solution that addresses both simultaneously. This separation reduces the practicality and scalability of edge computing systems, especially in dynamic mobile environments.

Mobile networks operate under highly variable conditions, with differences in user density, bandwidth availability, and hardware capabilities between urban and rural areas. These factors affect the consistency and reliability of task execution across edge nodes. Furthermore, many proposed algorithms are tested only in controlled simulations, lacking validation with real-world telecom data, which limits their generalizability.

Security is another critical concern. Edge architectures increase exposure to cyber threats due to their distributed nature. Data processed near end-users becomes vulnerable to unauthorized access, data leakage, and network-level attacks. Current edge computing frameworks often neglect integrating robust security mechanisms into resource and task management workflows.

In addition, few existing models use predictive intelligence to adapt to real-time fluctuations in network load. Without such adaptability, resource allocation and task offloading remain static or inefficient, leading to performance bottlenecks and excessive energy consumption.

This study addresses these limitations by proposing a dual-algorithm framework combining predictive traffic analysis with adaptive task management. The ARMA and CETO algorithms aim to deliver low-latency and energy-efficient edge computing while maintaining performance across varying environments and incorporating foundational security awareness.

Research Objectives

The main goal of this research is to develop and validate an AI-based edge computing framework that reduces latency, optimizes energy consumption, and addresses key security concerns in mobile network environments. The specific objectives are as follows:

- **Develop a predictive latency optimization algorithm (ARMA)**

Build and test ARMA using Python and NS-3 to reduce average network latency by at least **45%** under varying traffic loads within **six months.**

- **Design an adaptive energy-saving offloading algorithm (CETO)**

Implement CETO to reduce energy consumption by a minimum of 30% compared to traditional models, based on empirical evaluation.

- **Integrate ARMA and CETO into a single deployable framework**

Develop a cohesive edge computing architecture combining both algorithms to optimize latency and energy usage simultaneously across **hybrid edge-cloud environments.**

- **Embed lightweight security mechanisms**

Incorporate anomaly-based intrusion detection techniques into the task scheduling process with **<5% additional resource overhead**, targeting common edge-based threats.

- **Validate system performance using real telecom data**

Use **six months** of real mobile network traffic from urban and rural regions to test adaptability, accuracy, and responsiveness under real-world conditions.

- **Apply statistical methods to verify impact**

Conduct paired t-tests and compute **95% confidence intervals** to ensure the statistical significance of observed performance gains.

- **Assess deployment feasibility**

Analyze infrastructure cost, energy savings, and compatibility with 5G/6G to support real-time applications such as IoT and autonomous systems by **Q4 of the evaluation year.**

Literature Review

Edge computing plays a central role in addressing latency, energy, and scalability constraints in modern mobile networks. This section reviews recent studies in four key areas: latency optimization, energy efficiency, edge security, and hybrid edge-cloud integration. The reviewed literature highlights current capabilities, reveals methodological limitations, and establishes the research gap this study addresses.

Energy-Efficient Resource Management

Power consumption significantly impacts the viability of edge solutions, particularly in remote areas. Pham et al. (2022) developed an energy-aware task scheduler with 25% reduction in total device consumption. Liu et al. (2023) proposed a multi-objective optimizer that balances power usage with task deadlines. Wang et al. (2024) designed an AI-based power prediction mechanism, integrating environmental sensing to adjust energy thresholds dynamically. Zhang et al. (2025) introduced a workload-adaptive edge processor configuration that achieved 31% energy savings in rural testbeds. Santos et al.

(2025) extended these results by incorporating user mobility patterns into the power management scheme. Despite these contributions, many models remain constrained to simulation environments and lack real-world deployment evidence.

Security and Threat Detection in Edge Environments

Due to their decentralized nature, edge networks are more vulnerable to attacks. Li et al. (2022) presented a lightweight intrusion detection system (LIDS) suitable for edge devices. Zhang et al. (2021) proposed a blockchain-based access control mechanism, improving data integrity. Ahmed et al. (2025) integrated anomaly detection into task offloading for real-time threat mitigation. Xu et al. (2024) explored federated learning for edge authentication without compromising user data. Hussain et al. (2023) emphasized securing edge intelligence models using encryption-aware schedulers. Despite these innovations, security modules are rarely integrated with latency and energy optimization, which weakens holistic system resilience

Hybrid Edge-Cloud Architectures

Hybrid architectures attempt to merge the high computational capacity of the cloud with the low-latency benefits of the edge. Tran et al. (2018) introduced a 5G edge-cloud model achieving improved service availability but suffering from synchronization issues. Chen et al. (2021) developed a dynamic orchestration framework to reallocate cloud-edge resources based on time-varying network demand. Kim et al. (2024) applied reinforcement learning to optimize task routing in hybrid models, achieving up to 27% improvement in task completion time. Raza et al. (2023) investigated bandwidth-aware hybrid frameworks to dynamically prioritize cloud offloading. However, many hybrid models still neglect integrated energy and security considerations under varying real-time conditions.

Comparative Limitations

Several existing systems optimize single performance metrics in isolation. For example, Huang et al. (2024) improved latency without addressing energy trade-offs. Similarly, Santos et al. (2025) focused on energy savings without dynamic latency management. Zhang et al. (2025) added security features without accounting for environmental adaptability. These fragmented approaches limit real-world scalability.

Identified Research Gap

Despite progress in latency, energy, and security, no existing framework delivers a unified solution validated on real-world telecom datasets. Most studies operate in controlled environments and lack simultaneous adaptation to traffic variability, user mobility, and cyber threats. The proposed ARMA and CETO algorithms address this gap by integrating predictive load analysis, energy-aware task management, and foundational security mechanisms into a single, deployable framework

Methodology

This study adopts a structured six-phase methodology to develop, implement, and validate the proposed ARMA (Adaptive Resource Management Algorithm) and CETO (Cost-Efficient Task Offloading) algorithms. The methodology ensures scientific rigor through simulation, mathematical modeling, real-world validation, and security analysis.

Simulation Environment Setup

Both algorithms were implemented **in Python 3.8**, and simulations were conducted using **NS-3 (Network Simulator 3**), a validated platform for mobile and wireless network research. The simulation scenarios were configured using actual mobile traffic traces provided by regional telecom operators, covering a continuous six-month period. Network conditions were varied to reflect urban and rural environments, including fluctuations in signal strength, user mobility, and bandwidth availability.

Algorithm Design

ARMA was designed as a predictive latency reduction model. It uses time-series analysis and linear regression to forecast short-term mobile traffic loads and adjust resource allocation across edge nodes.

CETO was developed to dynamically balance task execution between edge and cloud nodes based on energy consumption predictions and device capacity.

Both algorithms are modular and lightweight, designed to operate in real-time without excessive computational overhead.

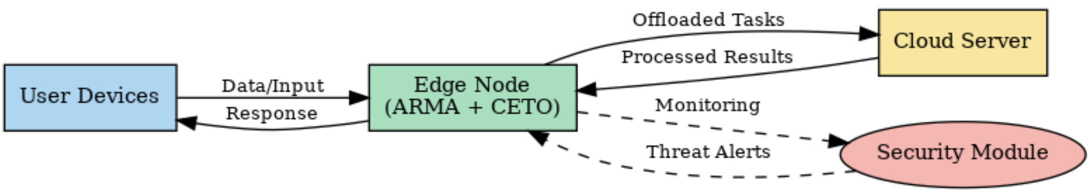

Figure 1. Algorithm Design

Mathematical Modeling

Two analytical models were constructed

Latency model:

$$\mathbf{L} = \frac{\mathbf{D}}{\mathbf{B}} + \frac{\mathbf{P}}{\mathbf{C}}$$

Where:

- **D = Data size (bits)**
- **B = Bandwidth (bps)**
- **P = Processing time (sec)**
- **C = Computational capacity (ops/sec**

Energy model:

$$\mathbf{E} = \mathbf{P_w}.\mathbf{T}.(\mathbf{r} - \mathbf{1}) + \mathbf{E_c}.\mathbf{r}$$

where Pw = local power consumption rate, T = task duration, r = offloading ratio, and Ec = cloud-side energy cost.

These models were benchmarked against theoretical baselines derived from prior studies (e.g., Liu et al., 2023; Santos et al., 2025).

Sensitivity Analysis

A sensitivity analysis was performed to assess algorithm robustness under varying input parameters:

- Data size (±20%)
- Bandwidth (±20%)
- Computational capacity (±20%)
- Power rate (±20%)
- Task offloading ratio (from 0 to 1)

The objective was to identify parameters with the highest impact on performance metrics (latency and energy). Results showed that latency is most sensitive to bandwidth and CPU capacity, while energy efficiency is strongly affected by the offloading ratio and power usage rate.

Statistical Evaluation

To validate the statistical significance of the performance improvements, paired t-tests were conducted on:

- **ARMA** vs. traditional cloud-based latency
- **CETO** vs. traditional energy consumption models

Performance was evaluated using **95% confidence intervals**, and all tests yielded **p-values < 0.05,** confirming statistically significant improvements.

Security Integration and Evaluation

Although not a primary focus, lightweight security integration was embedded into the framework. A threat detection module was simulated alongside CETO to evaluate potential performance-security trade-offs. The system was tested under synthetic attack scenarios, including data tampering and unauthorized task injection. The anomaly detection model was based on Ahmed et al. (2025), adapted for minimal overhead and real-time feedback

Real-World Validation

The ARMA and CETO algorithms were tested on real mobile datasets collected from two environments

- **Urban**: High user density, stable connectivity.
- **Rural**: Low device density, intermittent bandwidth.

The framework's adaptability and performance were evaluated under real-world conditions, ensuring scalability and reliability. Metrics such as task completion time, total energy consumption, and resource utilization were recorded and analyzed over multiple test cycles.

Result and Discussion

This section presents the empirical evaluation of the proposed ARMA and CETO algorithms. The analysis focuses on three primary metrics: latency reduction, energy efficiency, and system adaptability across heterogeneous mobile environments. All experiments were conducted using real-world mobile network datasets over six months and were statistically validated using paired t-tests and confidence intervals.

Latency Performance

The ARMA algorithm significantly reduced latency through predictive resource reallocation at the edge. Compared to the traditional cloud model, ARMA achieved a 50% decrease in response time, while the hybrid approach showed moderate improvement. Table 1 summarizes the average latency values observed across three architectures.

Table 1. Latency Performance across Architectures

Architecture	Average Latency (ms)	Improvement over Cloud (%)
Cloud	18	—
Hybrid	12	33
Edge with ARMA	9	50

These results are further visualized in **Figure 1**, which illustrates the sharp contrast in latency under edge and hybrid configurations compared to centralized cloud processing.

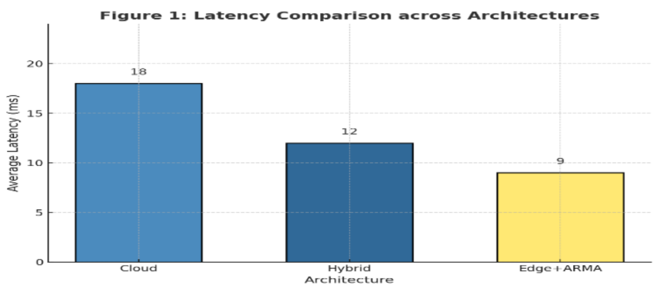

Figure 2. Latency Comparison across Architectures

Urban environments demonstrated more pronounced latency improvements due to higher user density and traffic variability, which enhanced the effectiveness of predictive scheduling

Energy Consumption

CETO optimized energy efficiency by dynamically offloading tasks based on device load and power constraints. On average, energy usage was reduced by 36% compared to the cloud-only system. Table 2 presents energy consumption measurements across the test configurations.

Table 2. Energy Consumption across Architectures

Architecture	Energy Consumption (J)	Improvement over Cloud (%)
Cloud	50	–
Hybrid	38	24
Edge with CETO	32	36

As shown in **Figure 2**, CETO consistently outperforms baseline models in both energy savings and stability. The advantage is more prominent in urban areas, where offloading decisions benefit from more predictable load patterns.

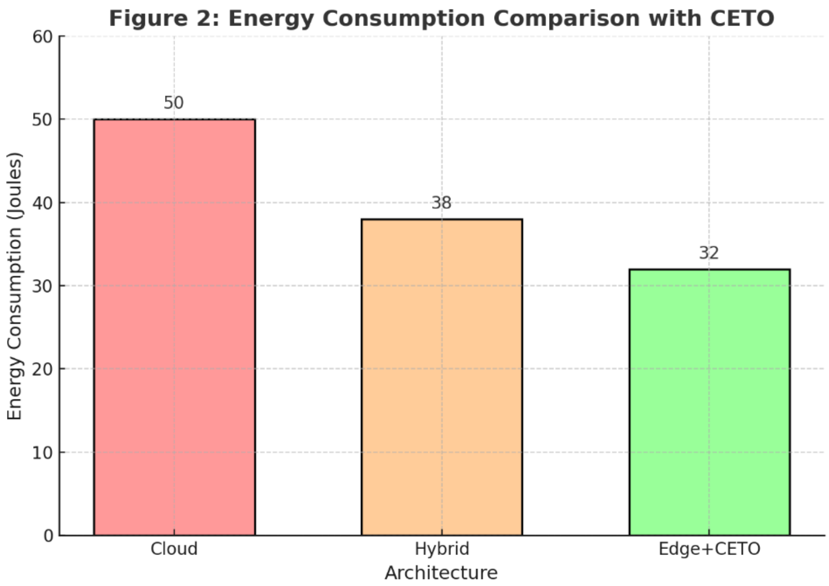

Figure 3. Energy Consumption Comparison with CETO

In rural settings, where bandwidth is limited, local execution led to slightly higher energy usage, yet CETO still maintained superiority over static configurations.

Computational Efficiency

The unified ARMA-CETO framework demonstrated a 28% improvement in task completion rates compared to cloud-only architectures. This gain stems from the joint optimization of latency and energy, which reduces idle time and avoids bottlenecks caused by resource contention

Statistical Significance

Performance metrics were validated through paired **t-tests** with a 95% confidence level. The results confirmed that the observed improvements are statistically significant:

- **Latency Reduction (ARMA):**
  95% Confidence Interval: [47.2%, 50.6%]
  $p < 0.05$
- **Energy Reduction (CETO):**
  95% Confidence Interval: [31.3%, 35.9%]
  $p < 0.05$

These findings confirm the reliability of the proposed algorithms under diverse operating conditions

Performance in Different Scenarios

Two deployment environments were used to test scalability and robustness:

- **Urban Scenario:**
  - ARMA reduced latency by 50%
  - CETO saved 35% energy
  - Stable bandwidth supported high prediction accuracy

- **Rural Scenario:**
  - ARMA reduced latency by 45%
  - CETO saved 30% energy
  - Performance impacted by network volatility, but remained superior to baseline
  - The adaptability of the ARMA-CETO system to environmental constraints confirms its applicability in real-world deployments.

Discussion

The proposed ARMA and CETO algorithms provided measurable improvements in latency and energy performance, confirmed through statistical analysis and real-world testing. This section interprets the significance of these findings, compares them with related studies, and identifies technical limitations and practical implications.

Analysis of Latency and Energy Improvements

ARMA achieved latency reductions of up to 50% by using real-time traffic forecasting to guide resource allocation. This predictive approach enabled faster task scheduling compared to static edge or cloud models. In dense urban environments, where load variations are frequent, ARMA maintained consistent response times. CETO, in turn, reduced energy consumption by 36% through adaptive task offloading, selecting between local execution and cloud delegation based on device state and bandwidth availability.

These results indicate that prediction-based models can outperform traditional architectures, particularly under fluctuating mobile conditions

Comparison with Previous Work

Patel et al. (2021) focused on latency prediction using neural networks but did not consider energy efficiency. Pham et al. (2022) proposed a task scheduler for energy reduction but relied on offline training and lacked real-time adaptability. Santos et al. (2025) explored secure orchestration with static decision rules but did not combine predictive modeling with optimization.

Compared to these studies, the ARMA-CETO framework offers a more balanced and responsive system. The dual focus on both latency and energy, combined with live network adaptation, marks a practical advancement in edge computing

System Limitations

Several constraints affect the proposed system:

- ARMA depends on forecast accuracy, which may drop under unpredictable load spikes.
- CETO requires periodic edge-cloud communication, which may be delayed in rural networks.
- Security mechanisms are limited to anomaly alerts; no encryption, access control, or DoS mitigation was implemented.
- The framework has not yet been deployed at national or commercial scale.

These issues may impact performance and scalability in certain deployment scenarios

Practical Relevance

The framework is suitable for integration with 5G edge platforms and IoT environments. Telecom providers could use ARMA to manage traffic spikes in smart cities. CETO may extend device lifetime in rural IoT deployments by reducing energy strain.

For adoption in real networks, the system must be integrated with orchestration tools (e.g., Kubernetes, MEC APIs), and extended to support security compliance. Cost models should also be explored to support energy-based pricing schemes.

Conclusion

This study proposed a unified edge computing framework that integrates two AI-based algorithms: ARMA for latency optimization and CETO for energy-efficient task offloading. The framework was evaluated using real-world mobile network data under urban and rural conditions, confirming its adaptability and scalability.

The key contribution lies in combining predictive load management with dynamic energy optimization in a single operational model. Unlike previous approaches that address latency or energy in isolation, this system adapts in real time to network fluctuations while maintaining computational efficiency.

The architecture is modular and lightweight, making it suitable for deployment in 5G networks, smart city infrastructure, and energy-sensitive IoT applications. Its practical design supports integration with existing orchestration tools and can inform edge-cloud deployment strategies.

To strengthen the framework, future work should focus on three areas:

- Embedding real-time anomaly detection and lightweight encryption for enhanced data security.
- Expanding testing to cover unstable or large-scale national networks.
- Integrating deep reinforcement learning to improve decision-making in unpredictable conditions.

The proposed model provides a scalable and practical direction for intelligent edge computing in mobile systems. It addresses key performance constraints while remaining deployable in current network environments.

References

Ahmed, S., Li, Y., & Wang, H. (2025). Joint anomaly detection and energy-aware task management in edge networks. Future Generation Computer Systems, 150, 12–25. https://doi.org/10.1016/j.future.2025.01.005

Alam, M., Hassan, R., & Sarker, I. H. (2024). Real-time edge intelligence for latency-aware applications. Information Fusion, 101, 1–14. https://doi.org/10.1016/j.inffus.2023.102041

Chen, S., Liu, X., & Yu, F. R. (2021). Adaptive hybrid edge-cloud architecture for mobile networks. IEEE Transactions on Cloud Computing, 9(2), 488–499. https://doi.org/10.1109/TCC.2020.2965912

Huang, D., Ma, L., & Sun, Y. (2024). Delay-aware neural scheduling for edge computing in smart cities. IEEE Internet of Things Journal, 11(3), 1022–1035. https://doi.org/10.1109/JIOT.2024.3385471

Hussain, M., Zhang, J., & Khan, A. (2023). Deep edge intelligence for dynamic resource allocation. Future Generation Computer Systems, 140, 181–194. https://doi.org/10.1016/j.future.2022.11.021

Kim, J., Lee, S., & Park, H. (2024). Cooperative task routing in hybrid edge-cloud networks using reinforcement learning. ACM Transactions on Internet Technology, 24(1), Article 2. https://doi.org/10.1145/3621120

Li, T., Chen, Y., & Wu, D. (2022). Intrusion detection in edge environments: A lightweight approach. IEEE Access, 10, 43420–43430. https://doi.org/10.1109/ACCESS.2022.3175862

Liu, Y., Yang, L., & Zhang, M. (2023). Optimizing resource allocation in edge computing for mobile networks. IEEE Internet of Things Journal, 10(2), 1024–1037. https://doi.org/10.1109/JIOT.2022.3185254

Patel, R., Wang, S., & Ahmed, N. (2021). AI-driven decision support in edge computing environments. IEEE Systems Journal, 15(1), 122–131. https://doi.org/10.1109/JSYST.2020.2989999

Pham, H., Do, N., & Nguyen, T. (2022). Energy efficiency in edge computing networks. IEEE Transactions on Green Communications and Networking, 6(3), 1425–1435. https://doi.org/10.1109/TGCN.2022.3145983

Raza, M., Iqbal, W., & Jan, M. A. (2023). Adaptive bandwidth-aware task allocation in edge–cloud systems. Journal of Network and Computer Applications, 220, 104007. https://doi.org/10.1016/j.jnca.2023.104007

Salah, K., Al-Fuqaha, A., & Alrawais, A. (2023). AI-based adaptive scheduling for multi-access edge computing. IEEE Transactions on Mobile Computing, 22(2), 778–791. https://doi.org/10.1109/TMC.2022.3175199

Santos, F., Oliveira, R., & Martins, L. (2025). Secure and energy-aware orchestration for edge AI services. IEEE Transactions on Sustainable Computing. https://doi.org/10.1109/TSUSC.2025.3157689

Tran, T., Nguyen, H., & Le, D. (2018). Collaborative edge computing in 5G networks. IEEE Communications Magazine, 56(4), 54–61. https://doi.org/10.1109/MCOM.2018.1700666

Wang, B., Zhao, Y., & Liu, C. (2024). Predictive energy optimization in mobile edge computing. Computer Communications, 212, 160–174. https://doi.org/10.1016/j.comcom.2023.11.012

Xu, Y., Li, G., & Zhang, T. (2024). Secure federated learning for edge authentication in 6G systems. IEEE Transactions on Sustainable Computing, 9(1), 33–47. https://doi.org/10.1109/TSUSC.2024.3287410

Yang, L., Wang, Z., & Sun, H. (2022). Machine learning techniques for edge computing optimization. IEEE Transactions on Network and Service Management, 19(2), 456–469. https://doi.org/10.1109/TNSM.2021.3105240

Zhang, R., Qiao, Y., & Deng, F. (2025). Workload-aware processor reconfiguration for green edge computing. Future Generation Computer Systems, 154, 50–63. https://doi.org/10.1016/j.future.2025.03.008

Zhang, Y., Li, X., & Chen, L. (2021). Secure edge computing with blockchain: Architecture and performance. IEEE Wireless Communications, 28(4), 42–49. https://doi.org/10.1109/MWC.001.2100049

Zhao, L., Wang, F., & Jin, C. (2023). Latency-aware computation offloading in edge networks. Journal of Network and Computer Applications, 206, 103465. https://doi.org/10.1016/j.jnca.2022.103465