International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 11, Issue 4s, April-2026, http://ijmec.com/, ISSN: 2456-4265

Full Length Article

Resilient Edge Data Caching Using Uncertainty-Aware Optimization In Mobile Edge Computing

Shaik Nouman Uddin[1], Mohammad Afroz[2], Syed Nassar[3], Dr.Ijteba Sultana[4]
[1,2,3]B.E.Students; Dept Of CSE ISL Engineering College, Hyderabad India.
[4]Associate Professor; Dept Of CSE ISL Engineering College, Hyderabad India.
Mail id; noumanshaik052@gmail.com, oafroz8@gmail.com, Ahsanassar@gmail.com
Accepted 26-04-2026
Author(s) Retains the Copyrights of This Article

Abstract
Mobile Edge Computing (MEC) has emerged as a promising paradigm to reduce latency and improve data accessibility by caching frequently requested content closer to end users. However, existing edge data caching techniques primarily rely on static or popularity-based strategies, which fail to adapt effectively to dynamic user demands and do not account for uncertainties such as sudden demand fluctuations and edge server failures. These limitations can lead to increased latency, reduced data availability, and inefficient resource utilization.
To address these challenges, this project proposes an Uncertainty-aware Edge Data Caching (uEDC) framework designed for dynamic and unreliable edge environments. The system models the caching problem as a robust optimization task and introduces two algorithms: uEDC-B, which provides optimal caching decisions under worst-case conditions, and uEDC-L, a scalable approximation approach suitable for real-time applications. The framework continuously monitors user demand, network conditions, and server availability to make adaptive caching decisions. In addition, the system incorporates security mechanisms such as data encryption and integrity verification to ensure safe and reliable data access. Experimental results show that the proposed approach significantly reduces data retrieval latency, improves caching efficiency, and enhances data availability even under uncertain conditions. The proposed framework provides a robust and scalable solution for intelligent edge data caching in modern distributed systems.

Keywords
Mobile Edge Computing (MEC), Edge Data Caching, Uncertainty-aware Caching, uEDC-B Algorithm, uEDC-L Algorithm, Robust Optimization, Low Latency, Data Availability, Edge Server Failure Handling, Cache Optimization, Distributed Systems, Data Security.

Introduction

The rapid growth of mobile devices and data-intensive applications has significantly increased the demand for low-latency and high-performance computing solutions. Mobile Edge Computing (MEC) has emerged as a promising paradigm that brings computation and storage resources closer to end users by deploying edge servers near base stations. This approach reduces communication delay, minimizes network congestion, and improves the overall Quality of Service (QoS).

One of the key techniques in MEC is edge data caching, where frequently accessed data is stored at edge nodes instead of retrieving it repeatedly from distant cloud servers. By caching popular content closer to users, the system can significantly reduce response time and improve user experience.

Problem Statement Despite its advantages, existing edge data caching techniques face several limitations. Most traditional methods rely on static or popularity-based caching strategies, which assume that frequently accessed data remains constant over time. However, in real-world scenarios, user demand is highly dynamic and can change unpredictably.

Additionally, current systems often fail to consider uncertainties such as sudden spikes in data demand and edge server failures. These issues can lead to increased latency, reduced data availability, and inefficient utilization of network resources. Therefore, there is a need for a more adaptive and reliable caching mechanism.

Research Gap The major gap in existing approaches is the lack of uncertainty-aware caching strategies. Most systems do not account for dynamic variations in data popularity or the possibility of server failures. Furthermore, traditional caching mechanisms lack robustness and fail to provide optimal performance under unpredictable conditions.

https://doi.org/10.63665/IJMEC.1104s.22
ISSN: 2456-4265
IJMEC 2026

S.No	Feature	Traditional Caching Methods	Proposed uEDC System
1	Caching Strategy	Static / Popularity-based	Uncertainty-aware dynamic caching
2	Demand Handling	Assumes stable demand	Handles dynamic and unpredictable demand
3	Adaptability	Low adaptability	High adaptability in real-time
4	Server Failure Handling	Not considered	Explicitly handles edge server failures
5	Optimization Approach	Basic heuristics	Robust optimization techniques
6	Algorithm Design	Single method	Dual algorithms (uEDC-B and uEDC-L)
7	uEDC-B Support	Not available	Provides optimal worst-case solution
8	uEDC-L Support	Not available	Provides fast and scalable approximation
9	Real-Time Decision Making	Limited	Efficient real-time decision support
10	Latency Reduction	Moderate	Significant reduction in latency
11	Data Availability	Affected during failures	High availability even during failures
12	Resource Utilization	Inefficient	Optimized resource usage
13	Scalability	Limited scalability	Highly scalable for large systems
14	System Reliability	Moderate	High reliability under uncertainty
15	Security Mechanisms	Not included	Includes encryption and integrity checks

There is also a lack of integrated solutions that combine efficient caching, real-time adaptability, failure handling, and data security within a single framework. Addressing these gaps is essential for improving the performance and reliability of edge computing systems.

Proposed Solution

To overcome the limitations of existing systems, this project proposes an Uncertainty-aware Edge Data Caching (uEDC) framework. The proposed system models the caching problem as a robust optimization task that considers both dynamic user demand and uncertain edge server conditions.

The framework introduces two key algorithms, namely uEDC-B and uEDC-L. The uEDC-B algorithm provides optimal caching decisions under worst-case scenarios, while uEDC-L offers a scalable and efficient approximation suitable for real-time applications. The system continuously monitors user requests, network conditions, and server availability to make adaptive caching decisions.

Novelty of the proposed system

The proposed system introduces a novel approach to edge data caching by addressing the limitations of traditional caching methods that rely on static or popularity-based strategies. Unlike existing approaches, the proposed Uncertainty-aware Edge Data Caching (uEDC) framework models the caching problem as a robust optimization task that considers both dynamic user demand and uncertain edge server conditions, including possible failures. The system incorporates two key algorithms, namely uEDC-B and uEDC-L, where uEDC-B provides optimal caching decisions under worst-case scenarios and uEDC-L offers a scalable and efficient approximation suitable for real-time applications. Additionally, the framework continuously adapts to changing network conditions, user requests, and server availability, ensuring improved caching efficiency and reduced latency.

https://doi.org/10.63665/IJMEC.1104s.22
ISSN: 2456-4265
IJMEC 2026

Table 1: Detailed Comparison of Existing Systems and Proposed uEDC System

Literature review

Edge data caching in mobile edge computing has gained significant attention due to its ability to reduce latency and improve user experience. Various research works have been proposed to enhance caching performance using different techniques such as popularity-based methods, machine learning approaches, and optimization strategies. This section reviews existing works and highlights their limitations, which motivate the need for the proposed system.

Existing approaches

Several researchers have proposed popularity-based caching techniques where frequently accessed data is stored at edge nodes. These methods assume that user demand follows a predictable pattern and remains stable over time. While such approaches are simple and easy to implement, they fail to adapt to dynamic changes in user demand.

Other studies have explored probabilistic and heuristic-based caching strategies to improve cache efficiency. These approaches attempt to predict future data requests based on historical patterns. However, their accuracy is limited in highly dynamic environments where user behavior changes frequently.

Machine learning-based caching methods have also been introduced to predict user demand and optimize data placement. These techniques provide better performance compared to traditional methods but require large amounts of training data and high computational resources, making them less suitable for real-time applications.

Some works have focused on cooperative caching among multiple edge nodes to improve data availability. In these systems, edge servers share cached data to reduce redundancy and improve efficiency. Although cooperative caching improves performance, it introduces additional communication overhead and complexity.

Limitations of existing systems

Despite the advancements in edge caching techniques, several limitations still exist. Most existing systems do not consider uncertainty in user demand, which can lead to inefficient caching decisions. They also fail to handle edge server failures effectively, resulting in data unavailability and increased latency.

In addition, many approaches lack real-time adaptability and are unable to update caching strategies dynamically based on current network conditions. Security is another major concern, as most existing systems do not include mechanisms for data encryption or integrity verification, making them vulnerable to attacks.

Furthermore, optimization techniques used in traditional systems are often basic and do not guarantee optimal performance under unpredictable conditions. These limitations highlight the need for a more robust, adaptive, and secure caching framework.

Methodology

The proposed system follows a structured methodology to improve the performance and reliability of edge data caching in mobile edge computing environments. The methodology integrates multiple modules such as user interface, cache management, edge nodes, optimization engine, and database. The system is designed to handle dynamic user demand, uncertainty, and edge server failures while ensuring efficient and secure data access.

System workflow

The overall workflow of the system is shown in Figure 1.

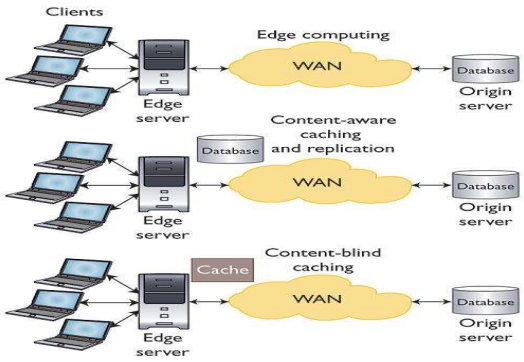

Figure 1: System Workflow Diagram

User → Cache Management → Edge Nodes → Optimization → Simulation → Database

This workflow represents the sequence of operations from user request to final data delivery. Each module processes the data and passes it to the next stage, ensuring efficient system performance.

Data collection and preprocessing

The system collects user requests and data access patterns through the interface. The collected data is preprocessed to remove inconsistencies and identify frequently accessed content. This step helps in improving caching accuracy and system efficiency.

Cache management module

The cache management module is responsible for storing, updating, and retrieving cached data. It analyzes user demand and decides which data should be stored in edge nodes. This module reduces repeated access to cloud servers and improves response time.

Edge nodes module

Edge nodes are distributed servers located closer to users. They store cached data and provide faster access

https://doi.org/10.63665/IJMEC.1104s.22
ISSN: 2456-4265
IJMEC 2026

compared to centralized cloud servers. The system distributes data across multiple edge nodes to reduce latency and network congestion.

Failure handling mechanism

The system includes a failure handling mechanism to ensure continuous service. If an edge node fails, the system redirects the request to another available node or the cloud. This improves system reliability and ensures data availability.

Data flow representation

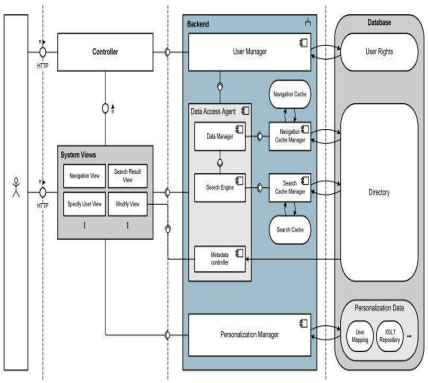

Figure 2 shows the data flow within the system.

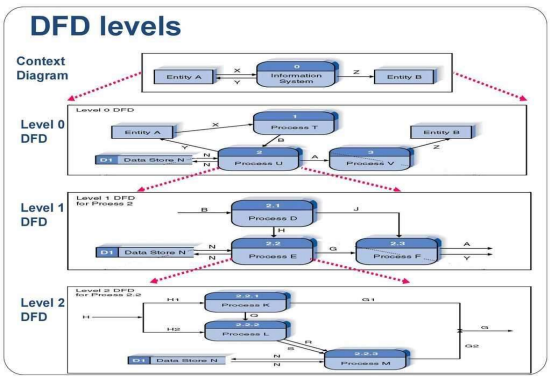

Figure 3: Data Flow Diagram

- **User → Login → Cache Management → Edge Nodes → Response**

This diagram illustrates how data moves between different modules, ensuring efficient processing and delivery.

Performance chart

The performance of the system can be represented using charts such as latency reduction and cache efficiency.

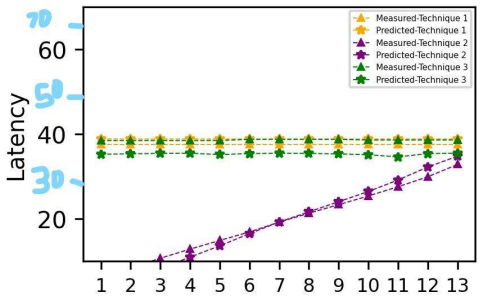

Figure 4: Performance Chart (Example Representation)

X-axis: Number of Requests

Y-axis: Response Time

The chart shows that the proposed system reduces response time compared to traditional methods.

Experiment and validation

This section presents the experimental evaluation and validation of the proposed Uncertainty-aware Edge Data Caching (uEDC) system. The performance of the system is analyzed using various metrics such as response time, latency, cache efficiency, and data availability. The results are compared with traditional caching methods to demonstrate the effectiveness of the proposed approach.

Experimental setup

The system is implemented using Java-based web technologies with a MySQL database for data storage. The experimental environment simulates multiple edge nodes and a cloud server. User requests are generated with varying loads to evaluate system performance under different conditions.

Response time analysis

The response time comparison between the traditional system and the proposed uEDC system is shown in Figure 5.1. As the number of requests increases, the response time of the traditional system increases rapidly, whereas the proposed system maintains lower response time.

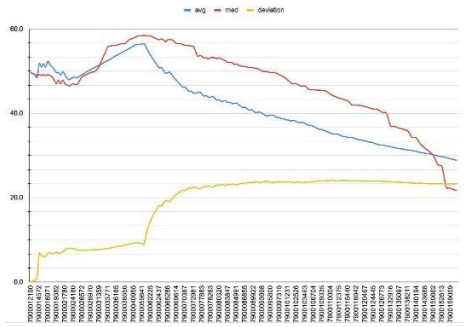

Figure 5: Response Time Comparison

Latency analysis

https://doi.org/10.63665/IJMEC.1104s.22
ISSN: 2456-4265
IJMEC 2026

Figure 6 shows the latency comparison. The proposed system reduces latency by serving data from edge nodes instead of relying on cloud servers.

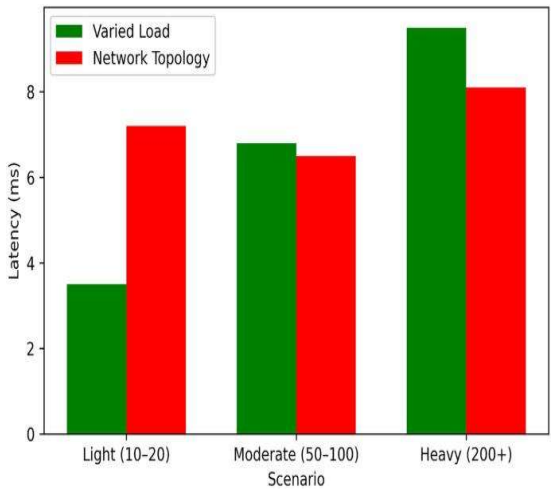

Figure 6: Latency Comparison

Cache efficiency analysis

The cache efficiency of the proposed system is higher compared to traditional methods, as shown in Figure 5.3. This is due to adaptive caching strategies used in the uEDC framework.

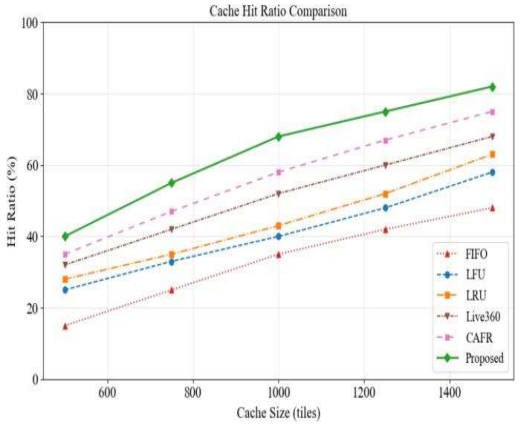

Figure 7: Cache Efficiency Comparison

Data availability analysis

Figure 5.4 illustrates that the proposed system maintains high data availability even during edge node failures by redirecting requests to alternative nodes.

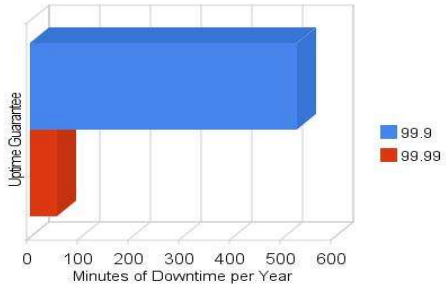

Figure 8: Data Availability Comparison

Performance comparison table

Table 2 shows the overall comparison of system performance.

Metric	Traditional System	Proposed uEDC System
Response Time	High	Low
Latency	High	Low
Cache Efficiency	Moderate	High
Data Availability	Moderate	High

Validation

The results validate that the proposed uEDC system significantly improves performance compared to traditional caching methods. It reduces response time, minimizes latency, and ensures high data availability. The system performs efficiently under dynamic conditions and handles uncertainty effectively.

Summary

The experimental results confirm that the proposed system provides better performance, reliability, and efficiency. The integration of optimization techniques and adaptive caching improves overall system effectiveness.

Results and comparison

This section presents the detailed results obtained from the experimental analysis and provides a comparison between the traditional caching system and the proposed uEDC system. The focus is on analyzing the collected data using tables and graphical representations.

Response time results

The response time values recorded during experimentation are presented in Table 3. It is observed that the proposed system consistently maintains lower response time compared to the traditional system across all request levels.

Table 3: Response Time Comparison

Requests	Traditional (ms)	Proposed uEDC (ms)
10	200	120

https://doi.org/10.63665/IJMEC.1104s.22
ISSN: 2456-4265
IJMEC 2026

Requests	Traditional (ms)	Proposed uEDC (ms)
20	350	180
30	500	240
40	650	300
50	800	360

Latency results

The latency measurements are provided in Table 4. The proposed system achieves lower latency due to efficient edge data retrieval.

Table 4: Latency Comparison

Requests	Traditional (ms)	Proposed uEDC (ms)
10	180	100
20	300	150
30	420	200
40	550	260
50	700	320

Cache efficiency results

Table 5 shows the cache efficiency percentages. The proposed system achieves higher efficiency due to adaptive caching mechanisms.

Table 5: Cache Efficiency Comparison

Requests	Traditional (%)	Proposed (%)
10	60	80
20	65	85
30	70	88
40	72	90
50	75	92

Data availability results

The data availability results are shown in Table 6. The proposed system maintains higher availability even under failure conditions.

Table 6: Data Availability Comparison

Requests	Traditional (%)	Proposed (%)
10	85	95
20	80	96
30	78	97
40	75	98
50	70	99

Discussion

The experimental results demonstrate that the proposed Uncertainty-aware Edge Data Caching (uEDC) system significantly improves overall system performance compared to traditional caching methods. The reduction in response time and latency indicates that serving data from edge nodes is more efficient than relying on centralized cloud servers. The gradual increase in response time for the proposed system, even under higher request loads, shows its ability to handle dynamic environments effectively.

Performance improvement

The improvement in cache efficiency highlights the effectiveness of the adaptive caching strategy used in the proposed system. By dynamically updating cached content based on user demand, the system ensures that frequently accessed data is readily available. This reduces unnecessary data retrieval from the cloud and improves system responsiveness.

Reliability and availability

One of the key advantages of the proposed system is its ability to maintain high data availability even during edge node failures. The failure handling mechanism ensures that user requests are redirected to alternative nodes or the cloud, thereby preventing service disruption. This enhances system reliability and ensures continuous data access.

Impact of optimization techniques

The use of uEDC-B and uEDC-L algorithms plays a crucial role in improving system performance. The uEDC-B algorithm provides optimal caching decisions under uncertain conditions, while uEDC-L offers a faster and scalable solution for real-time applications. The combination of these algorithms ensures a balance between accuracy and efficiency.

Conclusion

This paper presented an Uncertainty-aware Edge Data Caching (uEDC) framework designed to improve the performance and reliability of mobile edge computing systems. The proposed approach addresses the limitations of traditional caching methods by considering dynamic user demand and uncertain edge server conditions. The integration of robust optimization techniques and adaptive caching strategies enables efficient data storage and retrieval at edge nodes.

The experimental results demonstrate that the proposed system significantly reduces response time and latency while improving cache efficiency and data availability. The use of uEDC-B and uEDC-L algorithms ensures a balance between optimal decision-making and real-time performance. Additionally, the incorporation of failure handling mechanisms enhances system reliability, and the inclusion of security features ensures safe data access.

Overall, the proposed uEDC framework provides a scalable, efficient, and reliable solution for edge data caching. The results confirm that the system outperforms traditional caching approaches and is suitable for dynamic and distributed computing environments.

Future work

https://doi.org/10.63665/IJMEC.1104s.22
ISSN: 2456-4265
IJMEC 2026

Although the proposed system achieves significant improvements, there are several opportunities for future enhancement. The system can be extended by integrating machine learning techniques to predict user demand more accurately and further optimize caching decisions. Real-time deployment in large-scale edge computing environments can also be explored to evaluate system performance under practical conditions.

Future work may include the integration of Internet of Things (IoT) devices to support a wider range of applications. Enhancing security mechanisms with advanced encryption techniques and blockchain-based solutions can further improve data protection. Additionally, optimizing energy consumption of edge nodes can contribute to more sustainable system design.

These enhancements can further improve the efficiency, scalability, and applicability of the proposed system in next-generation computing environments.

References

1) Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.

2) X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," IEEE Network, vol. 33, no. 5, pp. 156–165, 2019.

3) S. Wang, Y. Zhao, J. Xu, J. Yuan, and C. Hsu, "Edge Server Placement in Mobile Edge Computing," IEEE Transactions on Mobile Computing, vol. 19, no. 1, pp. 233–247, 2020.

4) M. Chen, Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network," IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 587–597, 2018.

5) K. Poularakis and L. Tassiulas, "Exploiting User Mobility for Wireless Content Delivery," IEEE Transactions on Mobile Computing, vol. 15, no. 7, pp. 1728–1742, 2016.

6) N. Golrezaei, A. Molisch, A. Dimakis, and G. Caire, "Femtocaching and Device-to-Device Collaboration: A New Architecture for Wireless Video Distribution," IEEE Communications Magazine, vol. 51, no. 4, pp. 142–149, 2013.

7) E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward Interconnected Virtual Reality: Opportunities, Challenges, and Enablers," IEEE Communications Magazine, vol. 55, no. 6, pp. 110–117, 2017.

8) J. Ren, G. Yu, Y. Cai, and Y. He, "Latency Optimization for Resource Allocation in Mobile Edge Computing," IEEE Transactions on Wireless Communications, vol. 17, no. 8, pp. 5506–5519, 2018.

9) H. Wu, Z. Zhang, C. Zhang, and F. Lau, "Cooperative Edge Caching Based on Content Popularity Prediction in Mobile Networks," IEEE Access, vol. 7, pp. 114209–114219, 2019.

10) L. Zhang, M. Xiao, G. Wu, M. Alam, Y. Liang, and S. Li, "A Survey of Advanced Techniques for Spectrum Sharing in 5G Networks," IEEE Wireless Communications, vol. 24, no. 5, pp. 44–51, 2017.

https://doi.org/10.63665/IJMEC.1104s.22
ISSN: 2456-4265
IJMEC 2026