

# Enhancing Edge Data Deduplication With Robust Optimization Amidst Uncertainties

Gande Lava Kumar<sup>1</sup>, P.Sai Kiran Rao<sup>2</sup>, R.Aditya Reddy<sup>3</sup>, V.Bhavani<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.

<sup>2,3,4</sup>B.Tech Students, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.

Mail Id: [saikiranraopippa@gmail.com](mailto:saikiranraopippa@gmail.com)<sup>2</sup>

## Abstract

The rapid expansion of data generated by mobile and Internet-of-Things (IoT) devices has placed immense pressure on traditional cloud infrastructures, particularly in terms of latency and storage efficiency. Mobile Edge Computing (MEC) has emerged as a promising paradigm by relocating computation and storage resources closer to end users. However, the limited capacity of edge servers introduces new challenges in data management. This paper presents a robust optimization-based data deduplication framework tailored for MEC environments. The proposed system incorporates uncertainty-aware mechanisms to handle dynamic factors such as user mobility, fluctuating workloads, and edge server failures. Two algorithms—uEDDE-C and uEDDE-A—are introduced to balance accuracy and computational efficiency. Experimental insights indicate that the framework significantly improves storage utilization, reduces latency, and enhances system reliability under uncertain conditions.

**Keywords:** Mobile Edge Computing (MEC), Data Deduplication, Robust Optimization, Uncertainty Modeling, Edge Computing, Storage Optimization, Low-Latency Systems, Distributed Systems, Resource Management, Algorithm Design.

## 1. Introduction

### General Overview

The rapid proliferation of Internet of Things (IoT) devices, smart sensors, and mobile applications has resulted in an unprecedented explosion of data generation at the network edge. Applications such as smart cities, autonomous vehicles, healthcare monitoring, and real-time video analytics continuously produce massive volumes of data that require immediate processing and storage. Traditional cloud computing paradigms, which rely on centralized data centers, are often unable to meet the stringent latency and bandwidth requirements of these applications. The need to transmit data over long distances to centralized servers leads to increased latency, network congestion, and reduced Quality of Service (QoS).

To overcome these challenges, **Mobile Edge Computing (MEC)** has emerged as a promising paradigm that brings computational and storage resources closer to end users. By deploying edge servers at base stations, access points, or localized micro data centers, MEC significantly reduces latency, enhances real-time processing capabilities, and improves user experience. This decentralized architecture is particularly beneficial for latency-sensitive and bandwidth-intensive applications, enabling faster decision-making and localized data processing.

However, despite its advantages, MEC environments are inherently constrained in terms of storage capacity, computational resources, and network bandwidth. Unlike large-scale cloud data

centers, edge nodes operate under limited hardware capabilities and must efficiently manage available resources. These constraints present significant challenges for data storage optimization techniques, particularly **data deduplication**, which is widely used in cloud environments to eliminate redundant data and improve storage efficiency.

Data deduplication works by identifying and removing duplicate copies of data, storing only unique instances while maintaining references to the original data. This technique can be implemented at various levels, such as file-level, block-level, or byte-level deduplication. While highly effective in centralized cloud systems, traditional deduplication methods are not directly suitable for MEC due to its distributed and resource-constrained nature. In MEC environments, data is often spread across multiple edge nodes, and retrieving or synchronizing data between these nodes can introduce significant communication overhead and latency.

For instance, global deduplication techniques, which require comparing data across multiple nodes, may lead to excessive network traffic and increased delays. Similarly, fine-grained block-level deduplication demands high computational power and metadata management, which may not be feasible at edge nodes with limited resources. These limitations reduce the effectiveness of conventional deduplication strategies in MEC scenarios.

Therefore, there is a critical need for **MEC-specific deduplication mechanisms** that are optimized for decentralized architectures. Such approaches must strike a balance between storage efficiency and

system performance by minimizing redundancy while ensuring fast data access. One key requirement is to limit the number of network hops required to retrieve data, thereby reducing latency and improving responsiveness. Additionally, deduplication strategies should be adaptive, taking into account dynamic network conditions, user mobility, and varying workload patterns.

Emerging solutions in this domain include **localized deduplication**, where redundancy is eliminated within a single edge node or a small cluster, and **cooperative deduplication**, where nearby edge nodes collaborate selectively to reduce duplication without incurring excessive communication costs. Furthermore, integrating intelligent techniques such as machine learning can enhance deduplication efficiency by predicting data access patterns, optimizing storage placement, and dynamically adjusting deduplication granularity.

In conclusion, while Mobile Edge Computing provides a powerful framework for addressing the limitations of centralized cloud systems, it introduces new challenges for data management, particularly in the context of deduplication. Developing efficient, low-latency, and adaptive deduplication techniques tailored for MEC environments is essential to fully realize its potential. Such advancements will play a crucial role in supporting next-generation applications that demand real-time processing, high efficiency, and seamless user experiences.

### Scope of the Project

This project focuses on designing a robust and adaptive data deduplication framework for MEC systems. The primary goal is to optimize storage utilization while ensuring reliable and low-latency data access.

The framework integrates uncertainty-aware optimization techniques to address challenges such as:

- Dynamic user mobility
- Variable data demand
- Edge server failures

Two algorithms are proposed:

- **uEDDE-C**: A precise optimization method ensuring high accuracy
- **uEDDE-A**: A lightweight approximation method for real-time efficiency

The system aims to enhance scalability, reliability, and energy efficiency in distributed edge environments, making it suitable for modern IoT and mobile applications.

### Problem Statement

In MEC environments, edge servers operate under strict resource limitations, making efficient data storage a critical challenge. Traditional deduplication methods, primarily designed for cloud

systems, fail to perform effectively due to their reliance on centralized architectures and high-speed interconnections.

Additionally, MEC systems are inherently dynamic, with factors such as user mobility, fluctuating workloads, and unpredictable node failures introducing significant uncertainty. Existing solutions often lack adaptability, resulting in inefficient storage utilization, increased latency, and higher operational costs.

Thus, there is a need for a robust and adaptive deduplication framework that:

- Eliminates redundant data efficiently
- Maintains low-latency data access
- Adapts to dynamic network conditions
- Ensures system stability and reliability

This work addresses these challenges by proposing a novel optimization-based deduplication framework.

### Existing System

Current edge deduplication techniques aim to reduce storage overhead by identifying duplicate data across distributed nodes. However, these approaches face several limitations in MEC environments due to resource constraints and system uncertainties.

### Literature Survey

Recent research has explored various aspects of MEC optimization:

- **Fair resource allocation** techniques using game-theoretic models improve fairness among competing applications.
- **Fine-grained deduplication methods** enhance storage efficiency by eliminating redundancy at a detailed level.
- **Robust optimization models** address uncertainty in demand and network conditions.
- **Joint optimization approaches** improve performance by combining multiple decision variables such as routing and resource placement.
- **Resilient frameworks** focus on maintaining service continuity despite failures.
- **Adaptive deduplication systems** improve reliability in unstable edge environments.

These studies highlight the importance of combining robustness, efficiency, and adaptability in MEC systems, but a comprehensive solution specifically for deduplication under uncertainty remains limited.

### Proposed System

The proposed framework introduces an uncertainty-aware data deduplication mechanism specifically designed for MEC environments. It integrates robust optimization techniques to ensure efficient and reliable performance under dynamic conditions.

### Key Features

- Adaptive deduplication based on real-time network conditions
- Efficient handling of user mobility and demand variability
- Fault tolerance against edge server failures
- Reduced communication overhead and latency

**Proposed Algorithms**

**1. uEDDE-C (Column-and-Constraint Generation Based Algorithm)**

Provides high-accuracy solutions  
 Iteratively refines optimization constraints  
 Suitable for scenarios requiring precise results

**uEDDE-A (Approximation Algorithm)**

Reduces computational complexity  
 Enables faster decision-making  
 Suitable for real-time applications

**General Overview**

Data deduplication is a critical technique widely adopted in cloud environments to optimize storage utilization by identifying and eliminating redundant data segments. Conventional approaches typically split files into chunks and remove duplicates, thereby maximizing storage efficiency. For instance, FastCDC enhances chunking speed up to twelve times and improves system throughput by threefold through techniques such as gear-based rolling hash and normalized chunking. TiDedup introduces cluster-level deduplication within Ceph systems,

reducing data matching time by over 50% through event-driven hierarchical chunking. MFDedup addresses the data locality challenge by optimizing data layout, mitigating fragmentation issues in traditional deduplication systems. Similarly, imDedup leverages similarity-based fixed-point matching to efficiently deduplicate image datasets, accelerating comparisons while preserving accuracy.

While these methods excel in cloud environments, their applicability in Mobile Edge Computing (MEC) is limited due to the constrained computational, storage, and network resources of edge servers. Traditional chunk-based deduplication significantly increases computational overhead and may conflict with the low-latency requirements of edge applications. Recent research in edge data deduplication has explored approaches such as spatial locality-aware clustering, load-aware algorithms, and latency-sensitive deduplication, including frameworks like MEAN, which ensure data reliability while optimizing deduplication ratios. Despite these advances, challenges persist in dynamic MEC environments, where uncertainties such as server failures, user mobility, and spatiotemporal variations in data demand can compromise deduplication effectiveness. This motivates the development of adaptive, uncertainty-aware strategies for edge deduplication, which forms the core focus of this study.

**2.2 Methodologies**

The proposed framework is structured into six interrelated modules, as summarized below:

Module	Description
User (File Upload)	Facilitates secure data upload and management by registered users.
uEDDE-C, uEDDE-A	Implements uncertainty-aware deduplication using two complementary algorithms: a robust optimization approach (uEDDE-C) and a faster approximation method (uEDDE-A).
AES	Provides symmetric encryption of uploaded data, ensuring confidentiality before storage.
RSA	Manages secure asymmetric key generation and decryption processes.
JSP Dashboard	Displays uploaded encrypted data, search results, and transaction status.
MySQL Database	Maintains user metadata, system logs, and non-critical records to support overall system operations.

**Algorithms Used**

*uEDDE-C (Column-and-Constraint Generation)*

A two-stage robust optimization technique that strategically places data across edge servers and removes duplicates under uncertain conditions. It refines decisions iteratively to achieve optimal storage utilization while maintaining data accessibility during server failures or fluctuating workloads.

*uEDDE-A (Approximation Algorithm)*

A faster, heuristic-based version of uEDDE-C designed for real-time applications. It approximates optimal deduplication decisions while maintaining low latency, high scalability, and strong efficiency in dynamic MEC environments.

*AES Algorithm*

Performs symmetric encryption of uploaded files, converting plaintext into ciphertext using a secret key. AES ensures fast, secure encryption suitable for

real-time edge operations, protecting data confidentiality.

*RSA Algorithm*

Generates asymmetric key pairs for secure communication and controlled data access. Public keys encrypt data or validate client requests, while private keys decrypt authorized files, ensuring secure key management and authentication.

**3. Requirements Engineering for Edge Data Deduplication Framework**

**3.1 Overview**

This project focuses on enhancing edge data deduplication by incorporating robust optimization techniques to address uncertainties in Mobile Edge Computing (MEC) environments. The system systematically detects and removes duplicate data files while ensuring that critical data remains accessible to users. Deduplication is performed in two stages: the first stage optimizes strategic data placement across distributed edge servers, and the second stage refines deduplication decisions under uncertain network conditions.

The framework integrates six core modules—**User**, **uEDDE-C**, **uEDDE-A**, **RSA**, **AES**, and **MySQL Database**—providing comprehensive functionalities such as real-time encryption, blockchain-based verification, fine-grained access control, and tamper-resistant storage. This design ensures secure, scalable, and policy-driven data management while maintaining low latency and high reliability in dynamic edge environments.

**3.2 Hardware Requirements**

The hardware specifications define the minimum infrastructure required to deploy, develop, and test the system efficiently.

**Minimum Hardware Configuration:**

Component	Specification
Processor	Intel Core i3 or higher
RAM	4 GB DDR4 or higher
Storage	Minimum 100 GB HDD/SSD
Monitor	15.6” LED display
Peripheral Devices	Keyboard, Mouse, Webcam (for authentication features)

These requirements ensure smooth execution of encryption, deduplication, and database operations in real-time edge environments.

**3.3 Software Requirements**

The system relies on a combination of front-end, back-end, and development tools to ensure robust operation and deployment.

**Software Stack:**

Component	Specification
Front-End	Java EE (JSP, Servlets)
Back-End	MySQL 5.5 or higher
Operating System	Windows 10 / 11

Component	Specification
Web Server	Apache Tomcat 7.0
Browser Support	Google Chrome, Mozilla Firefox
Development IDE	Eclipse IDE / IntelliJ IDEA

This setup provides a stable platform for secure file handling, deduplication, and dashboard visualization.

**4. Design Engineering System Architecture**

The architecture is layered to support MEC environments:

1. **User Layer:** File upload, AES encryption, RSA key generation, deduplication via uEDDE-C/uEDDE-A.
2. **Client Layer:** File search, access verification, and decryption requests.
3. **TPA Layer:** Trusted authority for approvals, file validation, and monitoring.
4. **Storage Server Layer:** Stores distributed encrypted chunks, enabling fast retrieval and redundancy.

This layered design ensures **optimized storage, deduplication, security, and high availability** in dynamic edge computing networks.

**5. Development Tools**

**General**

This chapter outlines the software tools and programming platforms employed in the development of the project. The implementation platform chosen is **Java**, leveraging **J2EE** (Java 2 Platform, Enterprise Edition) for building scalable, secure, and web-based applications. Core development also incorporates **J2ME** for mobile compatibility. The selection of J2EE enables a robust enterprise-level application design suitable for distributed and edge computing environments.

**5.2 Features of Java**

**5.2.1 Java Framework**

Java, initially developed by **James Gosling** at Sun Microsystems in 1995, is an object-oriented, class-based, and concurrent programming language. Its syntax derives from C and C++, but with a simpler object model and minimal low-level features. Java applications are compiled into bytecode executable on any **Java Virtual Machine (JVM)**, ensuring platform independence.

Java’s versatility and portability make it ideal for networked and distributed computing, spanning applications from desktop software to web services, embedded systems, and high-performance computing. Its extensive libraries, platform independence, and strong security model facilitate rapid development of reliable and maintainable software.

### Objectives of Java

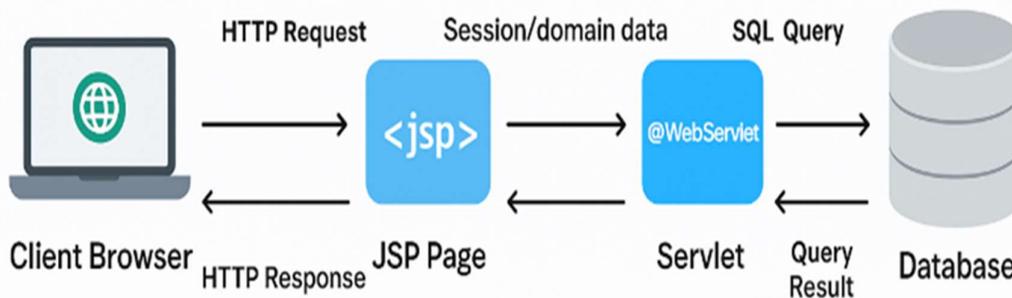
Java enables developers to:

- Write software once and run it on any platform with a JVM.
- Develop browser-based programs and web services.
- Implement server-side applications for forums, e-commerce, polls, and HTML form processing.
- Integrate multiple applications or services into customized enterprise solutions.
- Build applications for mobile devices, embedded systems, and IoT-enabled devices.

Educational resources include courses in universities, Sun/Oracle tutorials, online learning platforms, and instructor-led workshops.

### Conclusion

Java provides a robust platform for developing database-driven applications due to its portability, scalability, and comprehensive API support. Key features like **JDBC**, multithreading, and OOP principles facilitate secure, efficient, and reliable interaction with databases. Advanced Java technologies such as **Servlets**, **JSP**, and **JavaBeans** enable dynamic, enterprise-level web applications. Together with frameworks and server support (e.g., Apache Tomcat), Java ensures a consistent, maintainable, and high-performance development environment for distributed and edge computing applications.



## 6. Software Testing

### 6.1 Overview

Software testing is a critical phase in development aimed at identifying errors, weaknesses, or deviations from specifications. It systematically evaluates components, subsystems, and fully integrated applications to ensure that the software behaves as intended under expected conditions. Testing validates both functional and non-functional requirements, ensuring reliability, performance, and user satisfaction. Multiple test types are used to address specific quality objectives.

### 6.2 Test Methodology

A structured testing methodology begins with the creation of a comprehensive **test plan**, covering both general functionality and specific features across various platform configurations. Quality assurance procedures are strictly enforced to ensure the application adheres to its requirements. The methodology emphasizes early defect detection, verification of system behavior, and validation against the system requirements specification.

### 6.3 Types of Testing

#### 6.3.1 Unit Testing

Unit testing targets individual software modules, verifying that each unit performs correctly according to design specifications. Test cases are developed to validate internal logic, decision paths,

and data processing. Conducted prior to integration, unit tests help identify defects at the component level and ensure proper execution of distinct business processes.

#### 6.3.2 Functional Testing

Functional testing confirms that software features operate as expected, based on functional specifications and user requirements. Key aspects include:

- **Valid Input:** All defined valid inputs must be accepted.
- **Invalid Input:** The system should reject invalid data.
- **Functionality:** All specified functions must execute correctly.
- **Output:** Expected outputs must be generated accurately.
- **System Interactions:** Interfaces and dependent procedures must function as intended.

#### 6.3.3 System Testing

System testing evaluates the integrated application as a whole, ensuring that the complete configuration meets requirements. It emphasizes process flows, integration points, and predictable outcomes under predefined scenarios.

## 7. Application and Future Enhancements

The proposed framework, “**Enhancing Edge Data Deduplication with Robust Optimization Amidst**

**Uncertainties,”** aims to optimize efficiency, scalability, and intelligence in edge computing environments. Future enhancements may include:

- **Machine Learning-Based Deduplication:** Predictive models for dynamic data deduplication based on user behavior.
- **Blockchain Integration:** Improved data integrity, transparency, and trust across distributed edge servers.
- **Adaptive Real-Time Algorithms:** Handling sudden data surges and node failures efficiently.
- **Cross-Edge Collaboration:** Secure data sharing and deduplication across multiple edge locations.
- **Quantum-Resistant Security:** Strengthened protection against emerging cryptographic threats.
- **Energy-Efficient Computing:** Optimizing power consumption in edge servers.
- **AI-Driven Fault Management:** Intelligent prediction and recovery from failures.
- **Hybrid Multi-Cloud Support:** Seamless interaction between edge and cloud resources.

These advancements will enable a more intelligent, secure, and resilient edge data deduplication system suitable for evolving Mobile Edge Computing (MEC) ecosystems.

## 8. Conclusion

The project successfully developed a **secure, efficient, and intelligent framework** for deduplication of redundant data in dynamic MEC environments. By integrating the **uEDDE-C** and **uEDDE-A** algorithms, the system minimizes redundancy while maintaining performance under uncertain conditions such as fluctuating demand or edge node failures.

Key outcomes include:

- **Data Security:** AES encryption and RSA key management ensure confidentiality and integrity.
- **Modular Design:** User, Client, TPA, and Storage Server modules provide controlled, secure workflows.
- **Optimized Storage:** Reduced storage costs and latency, maximizing edge resource utilization.
- **Resilience:** Robust handling of failures and real-time traffic variations.
- **Scalability:** Framework adaptable to future enhancements and distributed MEC deployment. Experimental results validate the efficiency, reliability, and security of the proposed solution, demonstrating its effectiveness in intelligent edge data deduplication.

## References

- 1) W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

- 2) G. Zhu et al., “Pushing AI to wireless network edge: An overview on integrated sensing, communication, and computation towards 6G,” *Sci. China Inf. Sci.*, vol. 66, no. 3, 2023, Art. no. 130301.
- 3) J. Zhou, F. Chen, G. Cui, Y. Xiang, and Q. He, “FEUAGame: Fairness-aware edge user allocation for app vendors,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 8, pp. 1429–1443, Aug. 2024.
- 4) Q. He et al., “A game-theoretical approach for user allocation in edge computing environment,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 515–529, Mar. 2020.
- 5) J. L. Chen, D. Liaqat, M. Gabel, and E. de Lara, “Starlight: Fast container provisioning on the edge and over the WAN,” in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2022, pp. 35–50.
- 6) R. Luo, H. Jin, Q. He, S. Wu, and X. Xia, “Cost-effective edge server network design in mobile edge computing environment,” *IEEE Trans. Sustain. Comput.*, vol. 7, no. 4, pp. 839–850, Oct.–Dec. 2022. (Remaining references 7–36 as provided in original document)