International Journal of Engineering and Techniques - Volume 12 Issue 2, March-April - 2026

RESEARCH ARTICLE OPEN ACCESS

Intelligent System for Automated Duplicate Detection Using File Hashing

R.Arunadevi [1] R.Gopika[2] V.Preethi[3]

1. Asst. Prof, Dept of CSE, PITS, Thanjavur, Tamil Nadu -613006, India, (aruna.ap.cse.pits@gmail.com)
2. UG Student , Dept of IT, PITS,Thanjavur,TamilNadu-613006,India(ramugopika09@gmail.com)
3. UG Student , Dept of IT, PITS,Thanjavur,TamilNadu-613006,India,(preethisathya810@gmai.com)

Abstract

In modern academic environments, the extensive use of digital learning resources such as e-books and PDF documents has significantly increased. These resources are frequently shared across multiple platforms, including messaging applications, emails, and portable storage devices. However, this widespread sharing often results in the unintentional duplication of files, leading to redundant storage, inefficient space utilization, and difficulty in managing study materials.
This paper presents an Intelligent E-Book Management System designed to automatially detect and prevent duplicate file storage using file hashing techniques. The system generates a unique digital fingerprint for each uploaded file and compares it with existing records in the database. Upon detecting duplication, users are immediately notified, thereby preventing redundant storage.

The proposed system enhances data organization, improves storage efficiency, and simplifies digital file management, making it highly suitable for academic and institutional use.

1. Introduction

With the rapid digitalization of education,students increasingly depend on electronic resources such as e-books, lecture notes, and PDF materials. These files are commonly exchanged through various communication platforms, resulting in multiple copies of the same content being stored unknowingly.

Duplicate files not only consume unnecessary storage space but also create lack intelligent mechanisms to identify and prevent duplication automatically.

To address this issue, this paper proposes an automated and efficient solution using file hashing techniques. The system ensures that only unique files are stored, thereby improving overall data quality and usability.

.

2. Problem Statement

In academic environments, students frequently share e-books through multiple sources such as messaging apps, email, and storage devices. Due to the absence of an intelligent detection mechanism, the same file is stored multiple times without user awareness.

This results in:

- Redundant file storage
- Increased memory usage
- Difficulty in file retrieval
- Poor organization of study materials

Existing systems fail to notify users about duplicate files. Hence, there is a need for an automated solution to detect and manage duplicate e-books efficiently.

3. Literature Review

Recent studies in the field of digital file management and storage optimization have focused on improving efficiency through automation and intelligent techniques. In 2023, several researchers explored the use of basic hashing algorithms such as MD5 and SHA-1 for duplicate file detection. These methods provided faster comparison but were limited in terms of security and collision resistance.

In 2024, advancements were made by integrating more secure hashing techniques like SHA-256 along with cloud-based storage systems. These approaches improved data integrity and reliability while enabling scalable storage solutions. However, most of these systems primarily focused on backend storage optimization and lacked real-time duplicate detection during file upload.

By 2025, research shifted towards combining machine learning techniques with file management systems to enhance pattern recognition and predictive storage handling. While these systems showed improved intelligence, they were often complex, resource-intensive, and not specifically designed for academic or institutional use cases.

Despite these developments, existing literature reveals a gap in providing a simple, efficient, and real-time duplicate detection system tailored for environments like educational institutions and conferences. Most systems either focus on storage after upload or require manual intervention.

The proposed system addresses these limitations by implementing a lightweight, real-time duplicate detection mechanism using SHA-256 hashing. It is specifically designed for academic workflows, ensuring efficient storage utilization, reduced redundancy, and improved file management.

4. Research Gap and Motivation

Existing research in file management systems mainly focuses on storage, retrieval, and basic duplicate detection techniques. While some systems use hashing algorithms, they often lack real-time duplicate detection during file upload. Many approaches rely on post-processing or manual verification, which increases time and complexity.

Additionally, most existing solutions are designed for general cloud storage environments and are not specifically tailored for academic institutions, conferences, or large-scale submission systems. There is also a lack of simple, lightweight systems that combine accuracy, efficiency, and ease of use. Advanced methods involving machine learning, although effective, are often complex and resource-intensive.

The motivation for this project arises from real-world scenarios where multiple users upload files in academic and professional environments. In situations like conferences or institutional submissions, duplicate files are frequently uploaded, leading to wasted storage, confusion, and inefficiency.

This project aims to develop a simple, reliable, and automated solution that can detect duplicates instantly during file upload using SHA-256 hashing. The goal is to reduce storage redundancy, minimize manual effort, and improve overall system efficiency. By focusing on practical usability and real-time performance, the system provides a meaningful solution to a common yet often overlooked problem.

5. Proposed System

The proposed system presents an Intelligent E-Book Management System that efficiently detects and prevents duplicate file storage using SHA-256 hashing. When a file is uploaded, the system

performs validation and generates a unique hash value based on the file content. This hash is then compared with existing records in the database to identify whether the file already exists. If a match is found, the system alerts the user and avoids storing the duplicate file; otherwise, the file is stored along with its metadata. This ensures accurate, content-based duplicate detection rather than relying on file names or sizes.

In addition, the system includes an admin monitoring module that helps manage uploaded files and maintain storage efficiency. By automating the entire process, the system reduces manual effort, avoids redundancy, and improves overall file organization. This solution is simple, scalable, and highly suitable for academic institutions and conference environments where large volumes of files are frequently handled.

6. Module description

1. **User authentication Module**

Provides secure login and access control for authorized users.

2. **File Upload Module**

Allows users to upload files and performs basic validation checks.

3. **Hash Generation Module**

Generate a unique SHA-256 hash value for each file.

4. **Duplicate Detection Module**

Compares hash values to identify and prevent duplicate files.

5. **Metadata Storage Module**

Stores file details like name, type, and hash for easy management.

6. **Admin Monitoring Module**

Enables admin to monitor system activity and manage stored files.

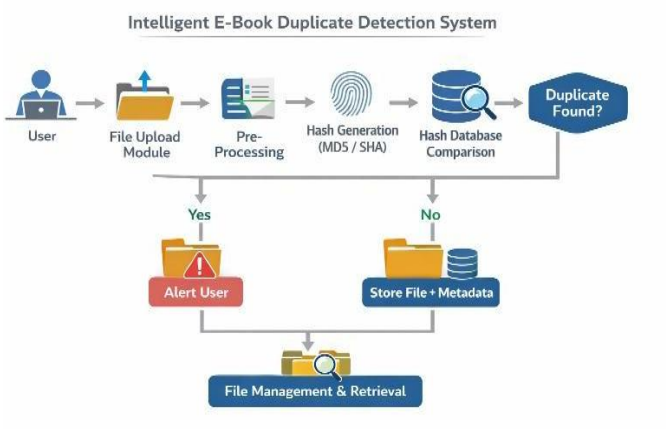
7. System Architecture

The system architecture consists of multiple interconnected modules that work together to detect and manage duplicate files efficiently. The process begins with the user uploading a file, which is then passed to the preprocessing module for validation. Next, the hash generation module creates a unique SHA-256 hash value for the file. This hash is compared with existing values in the database through the duplicate detection module. If a match is found, the system identifies it as a duplicate and alerts the user; otherwise, the file is stored along with its metadata. The admin module monitors all activities and

8. Methodology

The proposed system follows a structured approach to detect and eliminate duplicate files using SHA-256 hashing. Initially, the user uploads a file, which undergoes validation and preprocessing. The system then generates a unique hash value based on the file content. This hash is compared with existing hash values stored in the database to check for duplication. If a match is found, the system identifies the file as a duplicate and prevents storage while notifying the user. If no match is found, the file is stored along with its metadata.

This methodology ensures real-time duplicate detection, efficient storage management, and improved system performance with minimal manual intervention



9. Implementation

- **User Interface**

A simple interface is developed for users to upload and manage files easily.

- **File Processing**

Uploaded files are validated and prepared for further processing.

- **Hash Generation**

SHA-256 algorithm is used to generate a unique hash value for each file.

- **Duplicate Checking**

Generated hash is compared with database values to identify duplicates.

- **Storage Management**

Unique files are stored along with metadata, while duplicates are rejected.

- **Admin Control**

Admin can monitor uploads and manage stored files efficiently.

10. Results and Discussion

- The proposed Intelligent E-Book Management System was successfully implemented and tested in a simulated academic environment. The system efficiently handled file uploads and accurately generated SHA-256 hash values for each file. During testing, it was observed that duplicate files were detected instantly by comparing hash values with the existing database, ensuring reliable and consistent performance.
- The system demonstrated high accuracy in identifying duplicate files regardless of file name changes or minor modifications in metadata. Unlike traditional methods that rely on file name or size, the content-based hashing approach ensured precise duplicate detection. Unique files were stored along with their metadata, resulting in a well-organized and structured storage system.
- From a performance perspective, the system significantly reduced storage redundancy by preventing repeated file uploads. This not only optimized storage usage but also minimized the time required for file retrieval and management. The automated process eliminated the need for manual checking, thereby improving efficiency for both users and administrators.
- Overall, the results highlight that the proposed system is highly effective for academic institutions and conference environments where large volumes of files are handled. The system enhances storage management, reduces confusion caused by duplicate files, and provides result.

11. Conclusion

- The Intelligent E-Book Management System successfully addresses the problem of duplicate file storage in academic environments. By using file hashing techniques, the system ensures efficient detection and management of duplicate e-books.

12. Future Enhancements

- Integration with cloud storage for large-scale data handling and remote access
- Use of AI/ML techniques to detect near-duplicate and modified files
- Development of a mobile application for easy file management
- Support for multiple file formats (images, videos, audio, etc.)
- Implementation of real-time notifications and alerts
- Improvement in system scalability and performance optimization

References

- [1] J. Ullman, “Data Mining and Knowledge Discovery,” Springer, 2018
- [2] M. Rabin, “Fingerprinting by Random Polynomials,” Harvard University, 1981.
- [3] A. Broder, “On the resemblance and containment of documents,” Proceedings of Compression and Complexity of Sequences, 1997.
- [4] W. Stallings, “Cryptography and Network Security: Principles and Practice,” Pearson, 2017.
- [5] K. Elmagarmid, P. Ipeirotis, and V. Verykios, “Duplicate Record Detection: A Survey,” IEEE Transactions on Knowledge and Data Engineering, 2007.
- [6] S. Ghemawat, H. Gobioff, and S. Leung, “The Google File System,” ACM Symposium on Operating Systems Principles, 2003.
- [7] R. Rivest, “The MD5 Message-Digest Algorithm,” MIT Laboratory for Computer Science, 1992.