International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Smartdedup: An Intelligent And Secure Text De-Duplication Framework

[1]B Purushotham,[2]Uppari Rakesh,[3]Harijana Manohar, [4]Gandham Ajay Babu, [5]Kola Rohith Kumar

[1] Assistant Professor, Department of Computer Science & Engineering, Dr. K.V. Subba Reddy Institute of Technology

[2,3,4,5] B. Tech Students, Department of Computer Science & Engineering, Dr. K.V. Subba Reddy Institute of Technology

ABSTRACT

The exponential growth of digital textual data in cloud storage, enterprise systems, and online platforms has led to significant redundancy, increased storage costs, and security concerns. Traditional data de-duplication techniques focus mainly on exact matching and storage efficiency, often overlooking data security and privacy. This dissertation presents SmartDedup, an intelligent and secure text de-duplication framework that combines text similarity analysis with security mechanisms to identify and eliminate duplicate textual content safely. The proposed system uses preprocessing, hashing, similarity detection, and secure access control to ensure efficient storage utilization while preserving data confidentiality. SmartDedup aims to provide a scalable, privacy-aware, and reliable solution for managing large volumes of textual data.

Keywords: Text De-duplication, Secure Data Storage, Intelligent Data Processing, Data Redundancy Elimination, Hash-Based Matching, Similarity Detection, Natural Language Processing (NLP), Data Security, Privacy Preservation, Cloud Storage Optimization, Machine Learning-Based Classification, Content Fingerprinting.

I. INTRODUCTION

Textual data forms a major portion of digital information generated by businesses and individuals. Redundant storage of similar or identical text leads to inefficient resource utilization and higher operational costs. De-duplication techniques help eliminate redundancy by storing a single copy of repeated data. However, when applied to sensitive textual content, de-duplication must also address privacy and security concerns. SmartDedup integrates intelligent text analysis with secure data handling mechanisms, making it suitable for modern data-intensive and security-critical environments such as cloud storage and enterprise document management systems.

II. LITERATURE SURVEY

1. Title: Data De-Duplication Techniques for Storage Optimization

Authors: Meyer and Bolosky

Description:

This work discusses early data de-duplication techniques and their impact on reducing storage redundancy in large-scale systems.

2. Title: Secure De-Duplication of Encrypted Data

Authors: Bellare, Keelveedhi, and Ristenpart

Description:

The authors propose cryptographic methods to enable secure de-duplication while preserving data confidentiality.

3. Title: Near-Duplicate Detection in Large Text Collections

Authors: Broder

Description:

This study introduces shingling and hashing techniques for detecting near-duplicate text efficiently.

4. Title: Privacy-Preserving Data Storage Using Secure De-Duplication

Authors: Li, Yu, and Ren

Description:

This paper focuses on combining privacy preservation with de-duplication in cloud storage systems.

5. Title: Intelligent Text Processing for Large-Scale Data Management

Authors: Zhang and Liu

Description:

The authors highlight the role of intelligent text analysis in managing and optimizing large textual datasets.

III. EXISTING SYSTEM

Existing textual data de-duplication systems primarily rely on exact matching techniques such as hash comparison or checksum methods. These systems are effective for identifying identical files but fail to detect near-duplicate or semantically similar text. Additionally, most traditional systems do not provide adequate security features, making sensitive data vulnerable to unauthorized access and breaches.

IV. PROPOSED SYSTEM

The proposed SmartDedup framework introduces an intelligent and secure approach to textual data de-duplication. It combines text preprocessing, feature extraction, hashing, and similarity analysis to detect both exact and near-duplicate text. Security mechanisms such as encryption, authentication, and access control are integrated to protect sensitive data. The framework efficiently reduces storage redundancy while ensuring that only authorized users can access stored content.

V. SYSTEM ARCHITECTURE

The SmartDedup architecture is designed as a layered, modular, and secure framework that efficiently detects and removes redundant textual data while preserving privacy and integrity. The system is organized into multiple interconnected layers: Data Acquisition Layer, Preprocessing and Normalization Layer, Intelligent Deduplication Engine, Security and Encryption Module, Storage Management Layer, and Monitoring & Access Control Layer. Each layer performs a distinct responsibility, ensuring scalability, reliability, and security within cloud or enterprise environments.

At the first level, the Data Acquisition Layer collects textual data from multiple heterogeneous sources such as user uploads, databases, APIs, document repositories, email servers, or cloud storage platforms. This layer supports structured, semi-structured, and unstructured text formats including TXT, PDF, DOCX, logs, and web content. The ingestion module performs initial validation checks to ensure data integrity and format consistency before forwarding it to the preprocessing pipeline. A metadata tagging mechanism is also integrated to track file origin, timestamps, user identity, and document attributes for traceability and auditing purposes.

The Preprocessing and Normalization Layer prepares raw textual content for intelligent analysis. This layer applies Natural Language Processing techniques such as tokenization, stop-word removal, stemming or lemmatization, case normalization, punctuation filtering, and whitespace correction. It may also perform encoding normalization and noise removal to ensure uniform representation of content. Feature extraction methods like TF-IDF vectorization, n-gram modeling, or word embeddings are used to transform textual data into structured numerical representations. This step significantly improves the accuracy of similarity detection and duplicate identification by reducing irrelevant variations in text formatting.

The core component of the framework is the Intelligent Deduplication Engine. This module combines hash-based and similarity-based deduplication strategies to achieve both exact and near-duplicate detection. For exact duplicate detection, cryptographic hash functions such as SHA-256 generate unique fingerprints for each document. If two documents produce identical hashes, the system identifies them as exact duplicates. For near-duplicate detection, similarity comparison algorithms such as cosine similarity, Jaccard similarity, or MinHash-based locality-sensitive hashing (LSH) are applied on extracted features. A threshold-based decision mechanism determines whether documents are considered duplicates. The engine may also incorporate machine learning classifiers to enhance accuracy in identifying semantically similar texts, even when

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

wording differs significantly.

To ensure confidentiality and data protection, the Security and Encryption Module operates parallel to the deduplication engine. Before storage, documents undergo encryption using symmetric encryption techniques such as AES. Secure key management mechanisms ensure that only authorized users can access stored content. Additionally, secure hash storage and access verification protocols prevent unauthorized manipulation or tampering. This module enables deduplication to occur without compromising privacy, especially in cloud environments where multiple users share storage infrastructure.

The Storage Management Layer handles optimized storage allocation after duplicate detection. Instead of storing redundant copies, the system maintains a single instance of unique content and generates reference pointers for duplicate entries. This pointer-based indexing significantly reduces storage consumption and improves retrieval efficiency. Metadata indexing structures such as inverted indices or distributed hash tables support fast lookup operations. The architecture can be deployed over distributed cloud storage systems to ensure high availability and scalability, making it suitable for large-scale enterprise environments.

Finally, the Monitoring and Access Control Layer ensures secure and efficient system operation. Role-based access control (RBAC) mechanisms regulate user permissions for uploading, retrieving, or modifying documents. Audit logs record system activities for compliance and security analysis. Performance monitoring modules track metrics such as deduplication ratio, storage savings, processing time, and system throughput. Alerts and anomaly detection mechanisms can be integrated to detect suspicious behavior or unauthorized access attempts. Overall, the SmartDedup system architecture provides an intelligent, scalable, and secure solution for text deduplication. By integrating advanced NLP techniques, hash-based fingerprinting, machine learning-based similarity detection, encryption mechanisms, and optimized storage management, the framework ensures efficient redundancy elimination while maintaining strong data privacy and operational reliability.

Fig 5.1: Structure of the Proposed System

The given architecture diagram illustrates a secure cloud-based data deduplication framework that integrates convergent encryption and key management to ensure both storage efficiency and data confidentiality. In this system, multiple data owners (represented as Data Owner_u and Data Owner_v) interact with a centralized Cloud Server to upload (outsource) and download files. Before outsourcing a file, each data owner communicates with a dedicated Key Server (Key Server_i or Key Server_j) by sending a blind message/file. The term "blind" indicates that sensitive content is protected during communication, ensuring that even the key server cannot directly access the original file content. The key server then generates a Convergent Key along with a File Tag, which is returned to the respective data owner. The convergent key is derived from the file content itself, meaning identical files will produce identical keys, enabling duplicate detection without exposing the plaintext data.

When the data owner uploads the encrypted file to the Cloud Server, the server performs a deduplication check using the file tag. If the same file has already been stored by another user, the cloud avoids storing another copy and instead maintains a reference pointer, thereby saving storage space and bandwidth. If the file is new, it is securely stored in the cloud storage infrastructure. The Cloud Server thus performs two main operations: deduplication verification and secure storage management. When a

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

user requests to download a file, the cloud verifies authorization and provides the stored encrypted file, which the data owner can decrypt using the previously obtained convergent key. When a user requests to download a file, the cloud verifies authorization and provides the stored encrypted file, which the data owner can decrypt using the previously obtained convergent key. When a user requests to download a file, the cloud verifies authorization and provides the stored encrypted file, which the data owner can decrypt using the previously obtained convergent key. This architecture ensures efficient storage utilization by eliminating redundant copies while maintaining strong security through encryption, blind key generation, and controlled key distribution. Overall, the system balances data privacy, integrity, and storage optimization, making it suitable for secure cloud environments where multiple users may upload identical or similar files.

VI. IMPLEMENTATION

Fig 6.1: Uploading Dataset

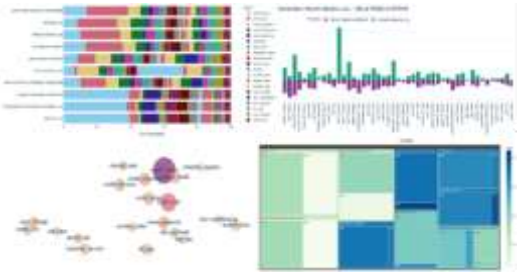

Fig 6.2: Feature Extraction

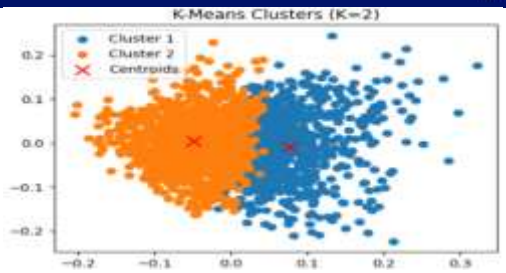

Fig 6.3: Duplication Detection and Cluster Analysis

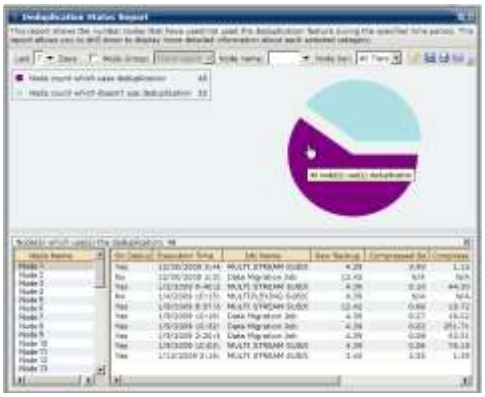

Fig 6.4: Secure Storage & De-duplication Summary Report

VII. CONCLUSION

SmartDedup presents an intelligent and secure approach to text de-duplication by combining effective text processing techniques with robust security mechanisms. The system efficiently identifies both exact and near-duplicate text documents using feature extraction and similarity analysis, thereby reducing redundant storage and improving data management efficiency. By employing secure hashing and encryption techniques, SmartDedup ensures confidentiality and integrity of user data throughout the deduplication process. The modular architecture enables reliable performance, scalability, and ease of maintenance. Overall, the proposed framework successfully addresses the challenges of storage optimization, data privacy, and accurate duplicate detection in modern text-based systems.

VIII. FUTURE SCOPE

In the future, SmartDedup can be enhanced by integrating advanced deep learning models such as

transformers to improve semantic similarity detection and handle paraphrased text more accurately. Multilingual text deduplication can be supported to extend usability across diverse language datasets. The framework can also be adapted for real-time deduplication in large-scale cloud environments using distributed processing techniques. Incorporating blockchain-based audit trails may further strengthen data integrity and transparency. Additionally, the system can be expanded to support other data types such as PDF, logs, and multimedia text, making it a comprehensive deduplication solution for enterprise applications.

IX. REFERENCES

[1]. M. Bellare, S. Keelveedhi, and T. Ristenpart, "DupLESS: Server-Aided Encryption for Deduplicated Storage," *IEEE Symposium on Security and Privacy*, 2013, pp. 179–194.
DOI: 10.1109/SP.2013.25

[2]. J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, "Secure Deduplication with Efficient and Reliable Convergent Key Management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1615–1625, 2014.
DOI: 10.1109/TPDS.2013.184

[3]. P. Anderson and L. Zhang, "Fast and Secure Laptop Backups with Encrypted De-duplication," *USENIX Annual Technical Conference*, 2010.
DOI: 10.5555/1855840.1855851

[4]. D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side Channels in Cloud Services: Deduplication in Cloud Storage," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 40–47, 2010.
DOI: 10.1109/MSP.2010.187

[5]. X. Yuan and C. Wang, "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage," *IEEE Transactions on Cloud Computing*, vol. 5, no. 2, pp. 218–230, 2017.
DOI: 10.1109/TCC.2015.2493505

[6]. J. Douceur et al., "Reclaiming Space from Duplicate Files in a Serverless Distributed File System," *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2002.
DOI: 10.1109/ICDCS.2002.1022240

[7]. S. Quinlan and S. Dorward, "Venti: A New Approach to Archival Storage," *USENIX Conference on File and Storage Technologies (FAST)*, 2002.
DOI: 10.5555/1083323.1083329

[8]. M. O. Rabin, "Fingerprinting by Random Polynomials," *Harvard University Technical Report*, 1981.
DOI: 10.1145/358669.358692

[9]. A. Broder, "On the Resemblance and Containment of Documents," *Compression and Complexity of Sequences*, 1997, pp. 21–29.
DOI: 10.1109/SEQUEN.1997.666900

[10]. G. Ateniese et al., "Proofs of Ownership in Remote Storage Systems," *ACM Conference on Computer and Communications Security (CCS)*, 2008.
DOI: 10.1145/1455770.1455799

[11]. K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.
DOI: 10.1109/MIC.2012.14

[12]. C. Dwork, "Differential Privacy," *International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.
DOI: 10.1007/11787006_1

[13]. P. Li, A. Shrivastava, J. Moore, and A. König, "Hashing Algorithms for Large-Scale Learning," *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.
DOI: 10.5555/2986459.2986575

[14]. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on

Large Clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008. DOI: 10.1145/1327452.1327492

[15]. K. Shilane, M. Huang, G. Wallace, and W. Hsu, "WAN-Optimized Replication of Backup Datasets Using Stream-Informed Delta Compression," *USENIX FAST*, 2012. DOI: 10.5555/2208461.2208466