Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 4, August 2025, pp. 2823~2836
ISSN: 2302-9285, DOI: 10.11591/eei.v14i4.9292

Optimizing cloud infrastructure efficiency through advanced multimedia data deduplication techniques

Mohd Hasan Mohiuddin[1], Latha Tamilselvan[2]
[1]Department of Computer Science and Engineering, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India
[2]Department of Information and Technology, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India

Article Info

Article history:

Received Sep 12, 2024
Revised Jan 21, 2025
Accepted Mar 9, 2025

Keywords:

Cloud computing
Cloud infrastructure efficiency
Cloud sim
Cloud storage optimization
Data deduplication
Hadoop distributed file system

ABSTRACT

Organizations worldwide commonly utilize cloud infrastructure to manage large volumes of data, making the optimization of storage crucial for enhancing cloud performance. One effective optimization technique is data deduplication, which identifies duplicate objects and ensures that only one copy of unique data is stored in the cloud. While several deduplication schemes currently exist, there is a pressing need to improve efficiency in cloud storage through innovative approaches. In this paper, we propose a new system model designed to facilitate an efficient deduplication process. Our algorithm, called deduplication in cloud infrastructure (DCI), offers a systematic and effective method for handling deduplication challenges related to redundant data storage. DCI focuses on hash generation, metadata comparison, and pointer-based deduplication, providing a comprehensive strategy for optimizing cloud storage resources and minimizing duplication. This ultimately enhances both the efficiency and cost-effectiveness of cloud-based data management. A simulation study using CloudSim and the Hadoop distributed file system (HDFS) simulator demonstrates that the proposed deduplication method is effective. Experimental results show that our algorithm outperforms many existing solutions, achieving the highest deduplication ratio of 6.7 and saving 85.09% of storage space due to its efficient deduplication approach. The proposed system can be used in cloud infrastructures for efficiency.

Corresponding Author:

Latha Tamilselvan
Department of Information and Technology
B.S. Abdur Rahman Crescent Institute of Science and Technology
Vandalur, Chennai – 600048, India
Email: latha.tamil@crescent.education

1. INTRODUCTION

Cloud computing enables sharing a large pool of resources to the public in a pay-per-use fashion. This technology has changed how companies in the real-world deal with storage and computing phenomena. With the emergence of cloud computing and other related technologies like big data, distributed computing, and the internet of things (IoT). Besides artificial intelligence (AI), the problems in the real-world application domains are being solved. Enterprises in the real world have found cloud computing to be a solution for storage and computing needs [1]. The rationale is that cloud storage is affordable, scalable, and available. Several services are rendered by the cloud, including platform as a service, infrastructure as a service, and software as a service. The cloud infrastructure is being used by various applications, including the ones that run on handheld devices [2]. As the cloud infrastructure stores large volumes of data, it is indispensable to explore different ways and means to leverage its performance.

A minor optimization in cloud infrastructure can add significant results because of cloud usage globally. Since big data is being maintained by cloud infrastructure, there is a need to investigate various methods to improve cloud infrastructure performance. One such well-known method to improve infrastructure optimization is deduplication [3]. Deduplication is a process of identifying duplicate objects in the cloud, eliminating duplicates, and ensuring only one unique copy of the data. With the emergence of cloud computing, people from all walks of life started using cloud infrastructure due to its tangible benefits. Commercial multimedia data providers have used cloud infrastructure for storage and management [4]. When multimedia objects are stored in the public cloud, duplicate objects may come from different users. In such cases, an object is stored in the cloud multiple times, wasting storage space and computing power. When there is redundant data, it may lead to cloud infrastructure inefficiency, including energy overhead. Venkatesan and Chitra [5] state that cloud storage is essential for sensitive data, but costs and security risks arise as data volumes grow. By removing duplication, the proposed ERCE-PF improves efficiency and security. The suggested approach reduces complexity by ensuring effective de-duplication and security with verified vital agreements. Khan *et al*. [6] by effectively transmitting patient data, the IoT-powered healthcare system lowers expenses and energy usage while enhancing treatment in various contexts.

Many existing approaches for deduplication in the cloud are found in the literature. Geetha [7] controlled file compression, de-duplication, node selection, load balancing, feedback control, and index name servers (INS) to simplify cloud storage. Unbalanced resources are the cause of cloud storage overflow. We provide hash-aware techniques and systems for effective data distribution [8]. unified data storage is encouraged by cloud computing and IoT. Effective de-duplication techniques like EFDS are required because SDN limits redundant data transfer. Said *et al*. [9], Because cloud computing unites many industries, safe data communication is necessary. We suggest risk-aware permission, ontology-based access restriction, and the use of pseudonyms [10]. Ample data access is made possible by integrating sensors and the cloud. Our proposal is an architecture for a sensor cloud that utilizes virtual sensors at the Infrastructure as a Service level [11]. The literature showed a need to improve the deduplication process. From the literature, it is evident that there are many existing methodologies for deduplication in cloud infrastructures (DCI). The current methods mainly focus on either file-based or block-level approaches to address this issue. However, the study of these existing methods highlights the need for a more comprehensive and hybrid approach that combines both file-level and block-level strategies. This would enhance flexibility and efficiency in the deduplication process. Therefore, the proposed methodology is designed to offer a novel approach that integrates both file-level and block-level methods, ultimately improving the efficiency of cloud infrastructure. Our contributions to this paper are as:

- Proposed an algorithm known as the DCI towards data deduplication.
- Built an application to evaluate the proposed algorithm for efficient deduplication.
- A simulation study was conducted using CloudSim and the Hadoop distributed file system (HDFS) simulator, and the results revealed that the proposed algorithm performs many existing algorithms with the highest deduplication ratio.

The remainder of the paper is structured as follows: section 2 reviews recent literature on data deduplication. Section 3 presents preliminary details that help in understanding the proposed methodology. Section 4 presents the proposed method for efficient data DCI. Section 5 presents the results of experiments with simulation studies using CloudSim and HDFS simulators. Section 6 discusses our work and provides the limitations of the study. Section 7 concludes our work and provides directions for the future scope of the research.

2. RELATED WORK

The section reviews the literature on existing methods for deduplication in the cloud. The proposed system aims to increase security and efficiency by strengthening data deduplication and access control across CSPs.

Sohani and Jain [12] impacted by resource provisioning issues in cloud computing. Through resource demand prediction, the PMHEFT algorithm enhances efficiency and load balancing. Ma *et al*. [13] encouraged efficiency, security, and practicality while thwarting collusion and duplicate fake assaults, the proposed server-side deduplication strategy in hybrid cloud architecture. Pugazhendi *et al*. [14] identified popular and unpopular files; the weight-based deduplication technology improves cloud storage by using less storage space. Mohan *et al*. [15] suggested speed-oriented de-duplication (POD) as a significant storage solution for cloud environments, emphasizing capacity reductions while enhancing I/O speed. Vijayalakshmi and Jayalakshmi [16], presented a deduplication framework that assesses big data and cloud computing about backup while protecting sensitive data. According to Gang and Wei [17] because of consistent manipulation indicators, such as contradictions and unsuitable content, Hindawi withdrew the piece. The literature revealed a need to improve the deduplication process.

Adhab and Hussien [18] created new methods for data deduplication and refining chunking algorithms to increase the effectiveness of cloud computing. Prajapati and Shah [19] concentrated on strengthening safe deduplication techniques, fixing significant management concerns, and boosting cloud storage systems' effectiveness. Sujatha and Raj [20] improved cloud storage efficiency and user memory use and compared deduplication strategies. Selvi and Sasirakha [21] improved integration and concentrated on creating uniform frameworks for efficiently administrating and archiving diverse IoT data. Neelamegam and Neelamegam and Marikkannu [22] enhanced data deduplication methods and dynamically optimized window size selection for better healthcare data management.

Kim *et al.* [23] improved deduplication methods to handle privacy issues, guaranteeing efficient and safe cloud services. Borade *et al.* [24] investigated safe multi-media data deduplication methods, emphasizing effective photo and video storage options. Arora and Vetrithangam [25] improved data deduplication methods and tackled performance, security, and storage efficiency issues in different applications. Manikyam and Devi [26] suggested the IDRPID-DD model for effective encryption, compression, and image deduplication, guaranteeing strong data security. Nagappan *et al.* [27] improved the logic and security of the suggested deduplication approach, investigated user privacy testing, and employed game theory. Manogar and Abirami [28] enhanced deduplication ratios for chunk identification and removing boundary shifting from the Smart Chunker algorithm. Table 1 presents a detailed summary of literature findings, showcasing the deduplication techniques, algorithms, datasets, and their identified limitations. This table aims to highlight the gaps in existing methodologies, such as the need for hybrid approaches and more comprehensive encryption mechanisms. The content is categorized to emphasize how each referenced work contributes to improving deduplication techniques while addressing specific limitations, such as granularity, scalability, and energy efficiency. The literature revealed a need to improve the deduplication process with hybrid approaches.

Table 1. Summary of literature findings

Ref	Technique	Algorithm	Dataset	Limitations
[2]	Deduplication	Proposed deduplication model	Locally available files dataset	The duplication process is to be improved, considering security concerns.
[5]	Enhanced randomized convergent encryption	de-duplication algorithm	Cloud database	Hybrid approaches are yet to be explored.
[6]	Energy-efficient de-duplication	Transmission of non-duplicated data to CH	Custom dataset	Energy-efficient approaches with compression are to be explored in the future.
[9]	De-duplication	Hash-table-based duplicate block identification and storage (HDBIS) algorithm	Custom dataset	This work's limitation is that it focuses on a single file to identify duplicate blocks.
[18]	Data deduplication	Data deduplication	Real-world datasets	A hybrid approach is desired to be developed in the future.
[19]	Convergent encryption	AES and blowfish algorithm	Local file system	The keyword search-based phenomenon will be improved in the future.
[20]	Deduplication	AE algorithm	Custom dataset	Supporting all kinds of data is left for the future scope of the research.
[22]	Health data deduplication	Window-size based chunking algorithm and advanced signature-based encryption (ASE)	Health care dataset	A dynamic moving window-based approach is to be explored in the future.
[25]	Deduplication	Upgraded chunking algorithms	Custom dataset	Granularity is missing in the current methodology.
[27]	Deduplication	Hybrid cloud storage with Diffie-Hellman algorithm	MS-SQL data set	A privacy-preserving approach will be incorporated into future work.

3. PRELIMINARIES

This section presents preliminary details that help readers understand the proposed methodology in this paper. It describes various fundamentals, such as the process of deduplication, different types of deduplication, and the usage of CloudSim.

3.1. Deduplication process

Deduplication is a process in which duplicate multimedia objects in storage infrastructure like the cloud are identified and removed from being stored in the cloud. When multiple copies of the same object are stored in the cloud, it leads to infrastructure inefficiency and wastage of computational power. As illustrated in Figure 1, only one copy is maintained for a particular object, while different pointers may exist, reflecting the exact copy for several users.

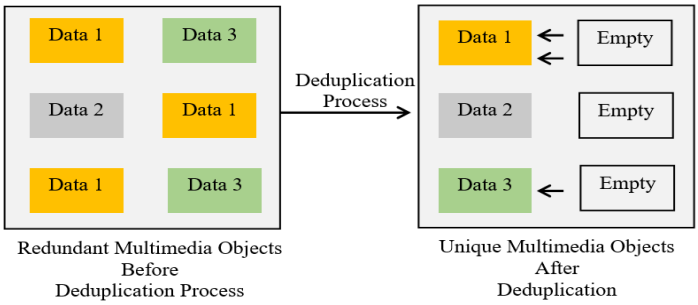

Figure 1. Process of deduplication

The deduplication process eliminates duplicate objects from the storage infrastructure, leading to infrastructure efficiency, energy efficiency, and performance leveraging in terms of meeting service level agreements (SLAs). With the process of deduplication, cloud resources are optimally utilized, improving the level of satisfaction for both consumers and service providers.

4. PROPOSED SYSTEM

This section presents the proposed methodology for the deduplication process toward efficient data storage management in the cloud.

4.1. Problem definition

Considering the number of duplicate multimedia objects coming from users across the globe, developing a deduplication methodology for eliminating duplicates and improving infrastructure efficiency is the problem considered. The methodology proposed in this paper for detection of duplicates and eliminating them contains novel approach picture consists of both file and block level mechanism. The deduplication method is provided in detail along with the underlying algorithm and its modus operandi.

4.2. Method for deduplication

Our system model is illustrated in Figure 2. It has mechanisms to perform the deduplication process and suggests a redundancy strategy if required to improve the quality of service (QoS). The present foundation of our system is an entire file hashing system for client-side Deduplication and a block-level approach. Thus, it is a model that improves efficiency in cloud infrastructure. Depending on the load of each deduplicator (proposed algorithm), the client performs the hashing process and connects to any of them. Using a comparison with the current hash values in the metadata server, the deduplicator determines whether duplicates have occurred. If a new hash value is found in classic deduplication systems, it is stored in the metadata server along with the file's logical path when posted to file servers. The file's reference count will be raised if it does exist. Depending on the system, each file may be kept in a fixed number of copies. To increase availability, files with many references could need additional copies. Some previous efforts added a degree of redundancy to deduplication systems to address this problem. Nevertheless, there are better indicators of redundancy level than determining redundancy by reference count, as specific files may be more important than others.

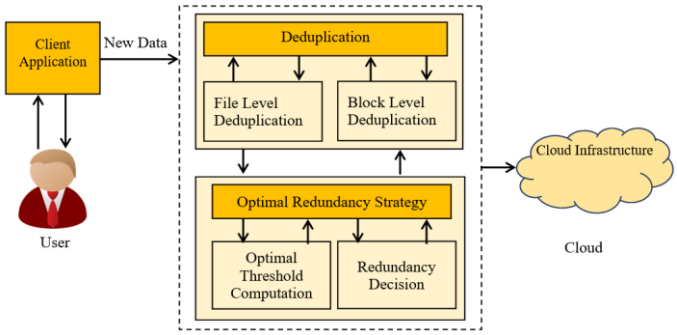

Figure 2. Overview of the deduplication model

Our suggestion is a deduplication solution that considers the cloud environment's QoS and dynamic nature, aiming to enhance availability without sacrificing storage economy. In our system model, the redundancy manager determines the appropriate degree of QoS and the number of references to determine the best number of copies for a file after recognizing the duplication. According to the fluctuating quantity of references, QoS level, and file demand, the number of copies is dynamically adjusted. Recalculation of the ideal number of copies by the redundancy management occurs when specific changes are tracked, such as a user deleting a file or an update to the file's QoS level. Below are the parts that make up the system.

Following the SHA-1 hashing process, clients use the load balancer to deliver a fingerprint, or hash value, to a deduplicator. By each deduplicator's current load, the load balancer reacts to requests from clients sent to any one of them. Deduplicator is intended to detect duplicates by contrasting them with the current hash values kept on file in the metadata server. In cloud storage, many file servers are used to store actual files and their copies, whereas a metadata server is used to store metadata. The redundancy manager is a part of determining the starting copy count and tracking the evolving QoS level.

4.3. Deduplication process

The process of data duplication is illustrated in Figure 3. When data arrives at the cloud infrastructure from any given client, it is essential to detect duplicates at the file level. Also, it duplicates at the block level to improve infrastructure efficiency. The approach required for the deduplication process is provided here. If the file that arrived has a duplicate in the cloud infrastructure, a reference of the same is added instead of saving file content. If there is no duplicate for a given file, the provided data is divided into several blocks, each used to generate a hash value. The generated hash values are compared with already stored hash values in the metadata server to identify duplicate files and blocks. Any block or file found as duplicate will not be stored in the cloud infrastructure, but its reference or pointer will be used for different users.

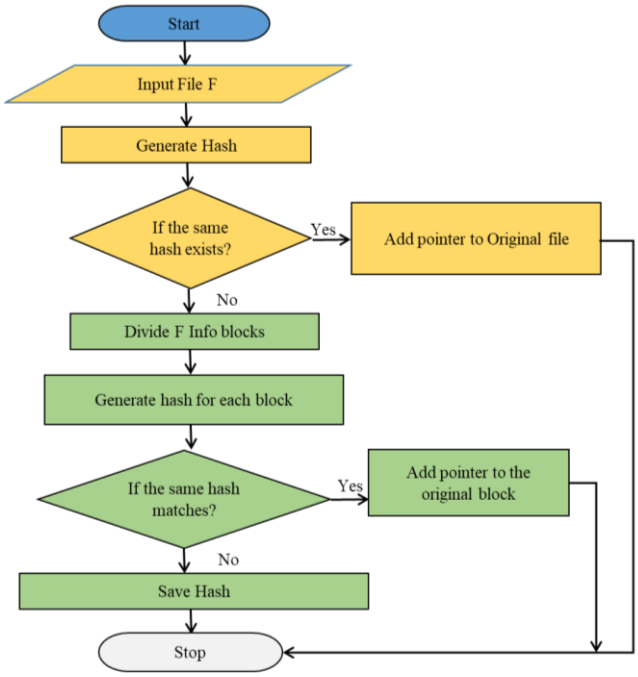

Figure 3. Deduplication process at file and block levels

In the streaming data that arrives at the cloud infrastructure, every data flow is verified to know whether it is an object that has been stored already. In the process, the fingerprint of the given block of data is computed, and then the fingerprint is matched with existing fingerprints in the metadata server. Based on the availability of a particular block in the cloud already, a decision is made whether to store it in the cloud infrastructure or make a reference or pointer to the existing object in the cloud. Maintaining indexing and metadata in the metadata server helps identify and eliminate duplicates in the cloud infrastructure.

Our suggested system model is being simulated using modified HDFS Simulation principles. As a metadata server, we establish one namenode; as file servers, we create five datanodes. In the metadata server, XML-formatted data is stored. Copying files is stored on file servers. We performed simulations for the upload, update, and delete events. The initial file upload to the system is called the upload event. The number

of copies of files already in the system and uploaded again will be adjusted for an update event based on the highest QoS. Users can delete their files using a delete file event; however, if another user refers to the same file, it won't be removed entirely from the system.

Once the client uploads a file, the deduplicator retrieves the hash value from it and uses it to search the metadata server for duplicates of that file. Should the file be fresh, it will be uploaded to the file server, and its updated metadata will be included in the system. By the QoS of the uploaded file, duplicates of the file will be generated. If the file already exists, its metadata will be modified, and the system could have to make duplicate copies of it or remove the original, depending on the file's maximum QoS value. To determine how many files the user wants to remove, the deduplicator counts the references to the same hash value. Each duplicate file copy will be erased if the hash is only mentioned once. However, reducing the number of file copies by the maximum value of QoS may be necessary if other files refer to the hash. In this case, the metadata will only be changed.

In the past few years, the growing importance of data storage requirements has encouraged the interest in the technique of deduplication. The studies carried out so far suggested different models and algorithms for performing the deduplication efficiently and utilized in their proposed models features of security, energy or hybrid approaches. This is because every approach adopts a different technique or uses different datasets and adopts certain limitations to address certain areas of the deduplication problem. One of the fundamental methods incorporates the use of a locally found deduplication model which is the basic of the models proposed, though security and other enhancements to the model are proposed for better protection. Enhanced randomized convergent encryption has also been used on cloud databases with a deduplication algorithm, but it does not explore hybrid approaches that could boost both efficiency and security. In another study, energy efficient deduplication is forwarded by sending non-duplicative data only to a central hub which can be improved further through the use of compression techniques for further energy savings. Hash-table-based block identification methods in the context of the duplicate storage application are effective but offer a single-file application leading to existing single-file applications offering little room for scalability. Several types of research indicate the need for further functionality. One example includes data deduplication for certain practical datasets, where the authors pointed out the potential for developing hybrid models with more versatility for future applications [18].

The application of convergent encryption with AES and blowfish algorithms in local file systems has a great potential, but it needs improvements in the keyword searching features [19]. A similar model lets users train AE on a self-made dataset but it currently does not offer support of many types of data making it possible to design more comprehensive deduplication methods [20]. Certain use cases like health data among others being able to utilize more advanced encryption while being able to apply window size based chunking techniques for de-duplication also highlight managing chunking techniques of appropriate dynamism for healthcare data UI only being unique for such needs [22]. Custom datasets have shown that enhanced chunking algorithms could make de-duplication easier but overall granularity is hampered leading to loss of accuracy [25]. Hybrid cloud storage and the Diffie-Hellman algorithm have also been combined on MS-SQL datasets with future plans focused on enhancing security features as additional studies to improve enhancing features of data security for infrastructure in the target space have been out [29]. All in all, the proposed methods illustrated the convergence in a trend supporting both file-level and block-level deduplication with very few exceptions.

4.4. Deduplication in cloud infrastructure

We proposed an algorithm known as the DCI for data deduplication. The DCI, in Algorithm 1, outlines a process for deduplicating files in a cloud storage environment. The algorithm begins by iterating through each file in a data stream and generating a fingerprint for the file. It then checks if the hash (fingerprint) of the file exists in the cloud's metadata. If the hash is found, a pointer to the original file is added, and the file is not saved to the cloud storage to prevent duplication. If the hash does not exist, the file is divided into blocks, and hashes are generated for each block. The algorithm then checks if any block hashes exist in the cloud metadata. If a match is found, a pointer to the original block is added, and the block is not saved to the cloud storage. The algorithm presents a two-step approach to DCI, aiming to minimize storage space by identifying and eliminating duplicate files and blocks. The algorithm effectively determines whether the data is stored by generating and comparing hashes of files and blocks with existing metadata in the cloud, avoiding redundancy. This approach optimizes storage resources and can contribute to cost savings in cloud environments by reducing the amount of data that needs to be stored. Additionally, the algorithm's use of pointers to the original files and blocks allows for efficient retrieval and access to the deduplicated data, maintaining data integrity and accessibility. Overall, the DCI algorithm offers a systematic and efficient method for DCI, addressing the challenge of redundant data storage. Its focus on hash generation, metadata comparison, and pointer-based deduplication provides a comprehensive strategy for optimizing cloud storage resources and minimizing duplication, ultimately enhancing the efficiency and cost-effectiveness of cloud-based data management.

Algorithm 1. DCI

Input: Data Stream D
Output: Data deduplication results

```
Begin
For each file F in Data Stream D
hash←GenerateFingerprint(F)
IF hash exists in cloud metadata Then
  Add a pointer to the original file
Do not save the file to cloud storage
  Else
blocks←DivideFile(F)
hashes←GenerateHashes(blocks)
    IF any hash in hashes exists in cloud metadata, Then
      Add a pointer to the original block
      Do not save the block to cloud storage
    Else
Save the blocks
    End If
  End If
End For
End
```

4.5. Optimal redundancy strategy

In cloud computing infrastructure, a file may be used by several users. In other words, a unique file may have several references, and that count may increase occasionally. Keeping one unique copy for all references may save memory but leads to a deteriorated QoS. To overcome this problem, we proposed an optimal redundancy strategy for maintaining several file copies despite the deduplication procedure. This will ensure the availability of data and faster access to data. Many existing systems used the level of redundancy as part of the deduplication procedure. Considering the level of redundancy based on several references pointed to the given file is not an ideal solution. We proposed a deduplication process integrated with an optimal redundancy strategy, considering several references pointing to a file and the associated QoS requirement. The consideration of both QOS and dynamicity has the potential to improve the availability of a given file.

As presented in Figure 4, the proposed optimal scheduling strategy computes the number of copies of a file to be maintained based on the dynamically increasing number of references and QoS requirements associated with the file. QOS requirement and several references point into a file or both dynamic. The proposed optimal scheduling strategy recomputes an optimal number of copies from time to time to determine the number of copies of a file to be maintained for sustained availability in cloud computing.

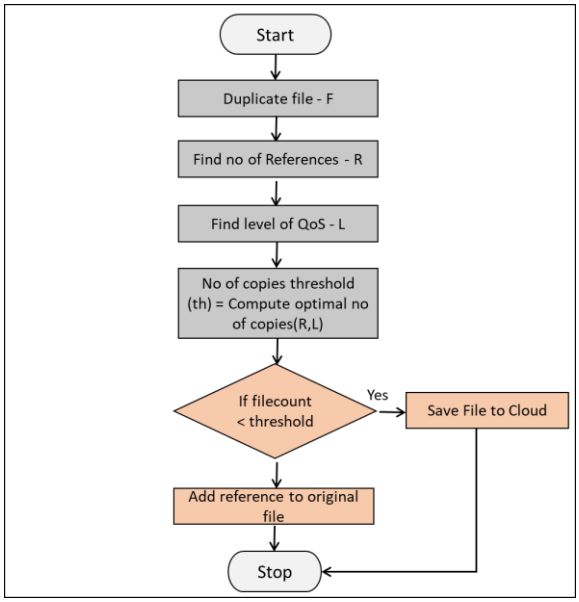

Figure 4. Strategic redundancy management

5. EXPERIMENTS AND RESULTS

Experiments are made with a Java-based prototype application that integrates the CloudSim framework and the HDFS simulator. The former is meant to simulate the deduplication process, while the latter is intended to simulate the storage infrastructure. Several experiments are made to make observations concerning deduplication. The proposed algorithms support both file and block-level deduplication. Combining both kinds of deduplication procedures will help improve cloud infrastructure performance. The proposed algorithm, DCI, is compared against many baseline deduplication methods. The existing methods used for comparison are ProbMinHash [29], SuperMinHash [30], and BagMinHash [31]. SuperMinHash, ProbMinHash, and BagMinHash are all algorithms designed to estimate the similarity between sets. SuperMinHash efficiently estimates Jaccard similarity, which measures how similar two sets are based on their intersection divided by their union. ProbMinHash focuses on the Jaccard similarity coefficient, while BagMinHash estimates similarity using "bag-of-words" or "bag-of-items" representations.

One of the critical observations is the duplication ratio (DR), which is computed by dividing total data before reduction (TDBR) by total data after reduction (TDAR), as expressed in (1). And the percentage of saved storage space (SSS), is computed by subtracting cumulative data (CD) from cumulative unique data (CUD), multiplied by 100, and dividing by cumulative data (CD) as expressed in (2).

$$DDR=\frac{TDBR}{TDAR} \quad (1)$$

$$SSS=\frac{(CD-CUD)*100}{CD} \quad (2)$$

Table 2 illustrates the experimental results obtained using sample 1 values, focusing on deduplication ratio and storage space saved across four experiments. The data demonstrates a consistent increase in deduplication ratio and percentage of saved storage space as cumulative data increases, showcasing the efficiency of the proposed DCI algorithm. For example, the deduplication ratio increases from 2.2 in experiment 1 to 3.0 in experiment 4, corresponding to storage savings from 54.54% to 66.66%.

Table 2. Experimental results using sample 1 values

Experiment	Cumulative data (TB)	Cumulative UniqueData (TB)	De-duplication ratio	Percentage of saved storage space
Experiment 1	1.1	0.5	2.2	54.54
Experiment 2	2.5	0.9	2.77	64
Experiment 3	3.5	1.2	2.91	65
Experiment 4	4.5	1.5	3	66.66

As presented in Figure 5, observations are made using sample 1 values. In the first experiment, cumulative actual data is 1.1 TB, while cumulative unique data is 0.5 TB. The percentage of saved storage space in the first experiment is 54.54 GB, in the second experiment 64 GB, in the third experiment 65 GB, and in the fourth experiment 66.66 GB, reflecting a gradual increase in the saved storage percentage. The results of an empirical study involving four experiments demonstrate that the proposed deduplication technique can consistently save storage space, achieving a deduplication ratio ranging from 2.2 to 3. This indicates a significant improvement in optimizing storage infrastructure in the cloud. The cumulative data used in the study across all four experiments ranged from 1.1 TB to 4.5 TB. Given that cloud computing infrastructure handles large volumes of data, even modest gains in storage efficiency can lead to substantial benefits. This performance enhancement not only optimizes the cloud infrastructure but also provides valuable advantages to cloud consumers.

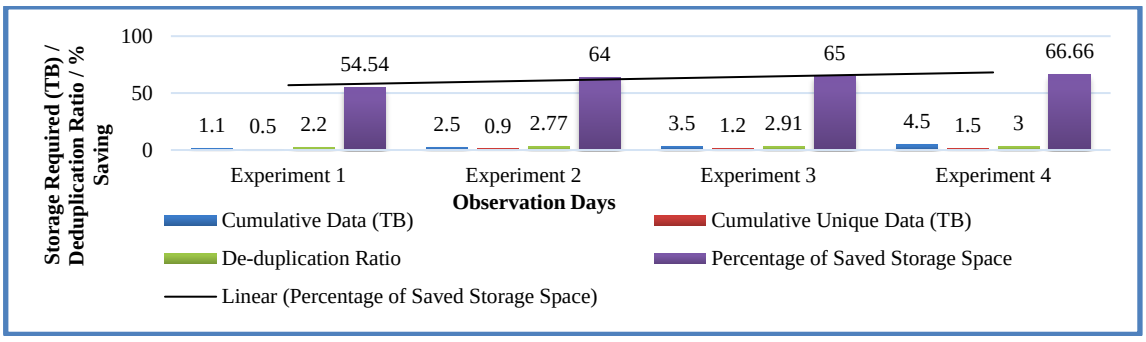

Figure 5. Percentage of storage saved after applying deduplication on sample 1 values

Table 3 presents the experimental results obtained using sample 2 values, focusing on deduplication ratio and percentage of saved storage space across eleven experiments. The table highlights the consistent improvement in deduplication efficiency as cumulative data increases. For instance, in experiment 1, with cumulative data of 1.2 TB and unique data of 0.8 TB, the deduplication ratio is 1.5, resulting in 33.33% saved storage space. As the cumulative data reaches 12 TB in experiment 11, the deduplication ratio rises significantly to 5.0, with 80% of storage space saved. This trend indicates the scalability and effectiveness of the proposed DCI algorithm when applied to larger datasets. The deduplication process not only eliminates redundant data effectively but also maximizes storage efficiency. These results validate the capability of the DCI algorithm to optimize cloud infrastructure resources, even in scenarios with diverse data volumes and characteristics.

Table 3. Experimental result using sample 2 values

Experiment	Cumulative data (TB)	Cumulative unique data (TB)	De-duplication ratio	Percentage of saved storage space
Experiment 1	1.2	0.8	1.5	33.33
Experiment 2	2.2	0.9	2.44	59.09
Experiment 3	3.2	1.1	2.90	65.62
Experiment 4	4.2	1.3	3.23	69.04
Experiment 5	5.2	1.5	3.46	71.15
Experiment 6	6.2	1.7	3.64	72.58
Experiment 7	7.2	1.8	4	75
Experiment 8	8.2	2	4.1	75.609
Experiment 9	9.2	2.1	4.38	77.17
Experiment 10	10.2	2.3	4.43	77.45
Experiment 11	12	2.4	5	80

As presented in Figure 6, observations are made using sample 2 values. In the first experiment, cumulative actual data is 1.2 TB, while cumulative unique data is 0.8 TB. The percentage of saved storage space in the first experiment is 33.33 GB; in the second experiment, 59.09 GB; in the third experiment, 65.62 GB; and in the last experiment, 80 GB, reflecting a gradual increase in the saved storage percentage.

The experimental results with various datasets indicate that the proposed deduplication methodology performs well across different tests. A total of 11 experiments were conducted, focusing on the deduplication ratio and the percentage of storage space saved. Several key observations emerged from these experiments. Notably, as the cumulative data in the cloud infrastructure increased, the proposed deduplication method demonstrated an enhanced ability to save storage space by effectively detecting and eliminating duplicate data. The deduplication ratio consistently increased across all experiments, ranging from 1.5 to 5.

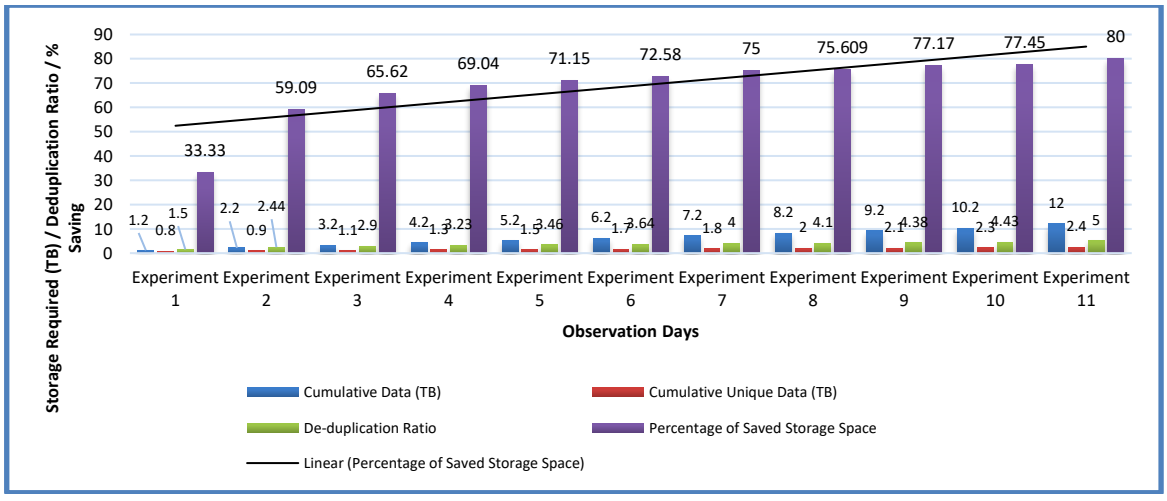

Figure 6. Percentage of storage saved after applying deduplication on sample 2 values

Table 4 showcases the experimental results using sample 3 values, detailing the deduplication ratio and percentage of saved storage space across 16 experiments. The results demonstrate a clear progression in the efficiency of the proposed DCI algorithm as cumulative data increases. In experiment 1, with cumulative

data of 1 TB and unique data of 0.8 TB, the deduplication ratio is 1.25, achieving a 20% storage space savings. By experiment 16, with cumulative data of 20.8 TB and unique data of 3.1 TB, the deduplication ratio significantly improves to 6.7, with an impressive 85.09% storage space savings. This table highlights the capability of the DCI algorithm to achieve high levels of data deduplication and resource optimization, even with increasing dataset sizes. The steady growth in the deduplication ratio and storage space savings reflects the robustness and scalability of the algorithm. These results underscore the practicality of the DCI approach for real-world cloud storage environments, where large-scale data management and optimization are critical.

Table 4. Experimental result using sample values

Experiment	Cumulative data (TB)	Cumulative unique data (TB)	De-duplication ratio	Percentage of saved storage space
Experiment 1	1	0.8	1.25	20
Experiment 2	2	0.9	2.2	55
Experiment 3	3	1	3	66.66
Experiment 4	4	1.2	3.3	70
Experiment 5	5.2	1.4	3.7	73.07
Experiment 6	6.5	1.6	4.06	75.38
Experiment 7	8	1.9	4.2	76.25
Experiment 8	9.5	2	4.75	78.94
Experiment 9	11	2.2	5	80
Experiment 10	12.5	2.3	5.4	81.6
Experiment 11	14	2.5	5.6	82.14
Experiment 12	15.2	2.6	5.8	82.89
Experiment 13	16.2	2.7	6	83.33
Experiment 14	18	2.9	6.2	83.88
Experiment 15	19.8	3	6.6	84.84
Experiment 16	20.8	3.1	6.7	85.09

As presented in Figure 7, observations are made using sample 3 values. In the first experiment, cumulative actual data is 1 TB, while the cumulative unique data is 0.8 TB. The percentage of saved storage space in the first experiment is 20 GB, in the second experiment is 55 GB, in the third experiment is 66.66 GB, and in the last experiment, it is 85.09 GB, reflecting a gradual increase in the saved storage percentage. Using the third data set, all the experiments made with the proposed deduplication technique consistently showed performance advantages. The experiments were conducted using a variety of inputs, and the observations were focused on the deduplication ratio achieved and the percentage of storage space saved. The findings from all experiments indicate that when the cumulative data reached 1 TB, the deduplication ratio achieved was 1.25, with a storage space savings of 20%. In contrast, during the last experiment, when the cumulative data amounted to 20.8 TB, the deduplication ratio increased significantly to 6.7, resulting in the highest percentage of storage space saved at 85.09%. These observations are based on the data sets and the underlying duplicate objects that were identified and subjected to deduplication.

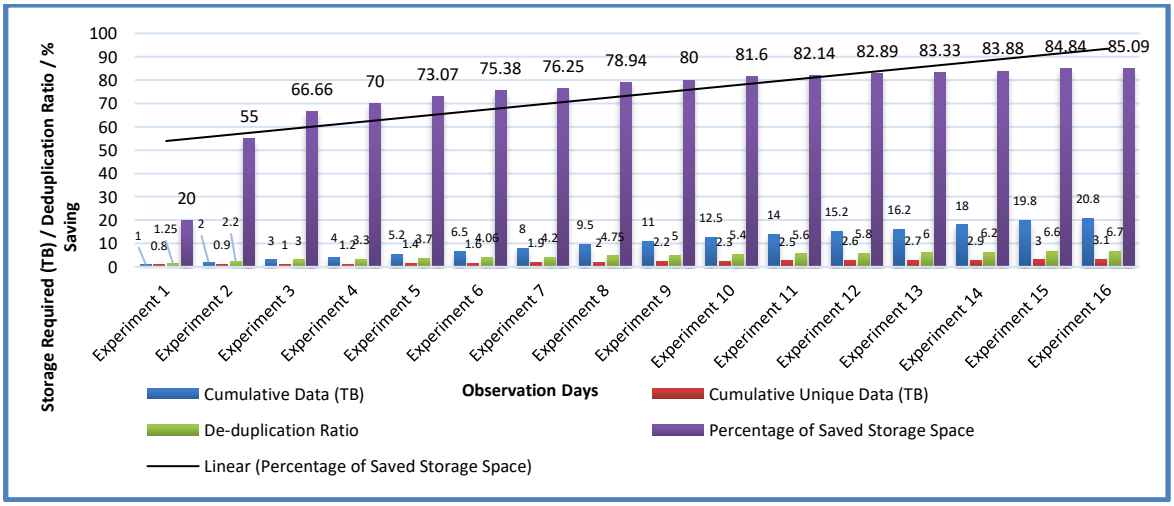

Figure 7. Percentage of storage saved after applying deduplication on sample 3 values

Table 5 provides a performance comparison of the proposed DCI algorithm against state-of-the-art methods such as SuperMinHash, ProbMinHash, and BagMinHash. Metrics include cumulative data, cumulative unique data, deduplication ratio, and percentage of saved storage space. The table highlights the superior performance of the DCI algorithm, which achieves the highest deduplication ratio (6.7) and storage space savings (85.09%), significantly outperforming existing methods.

Table 5. Performance comparison

Deduplication method	Cumulated data (TB)	Cumulated unique data (TB)	De-dup ratio	Percentage of saved storage space
SuperMinHash	20.8	7.4	2.8108	64.4230
ProbMinHash	20.8	6.9	3.0144	66.82692
BagMinHash	20.8	8.5	2.4470	59.13462
DCI (proposed)	20.8	3.1	6.7096	85.0961

Figure 8 presents a performance comparison of four deduplication methods: SuperMinHash, ProbMinHash, BagMinHash, and DCI (proposed). The compared metrics include cumulated data (TB), cumulated unique data (TB), deduplication (de-dup) ratio, and the percentage of saved storage space. All methods have the same cumulated data of 20.8 TB. For cumulated unique data, the values for SuperMinHash, ProbMinHash, BagMinHash, and DCI are 7.4 TB, 6.9 TB, 8.5 TB, and 3.1 TB, respectively. The de-dup ratio is highest for DCI at 6.709677419, followed by BagMinHash at 2.447088324, ProbMinHash at 3.014492754, and SuperMinHash at 2.810810811. Regarding the percentage of saved storage space, DCI leads with 85.9051385%, followed by BagMinHash with 59.31461538%, ProbMinHash with 66.82092908%, and SuperMinHash with 64.2307692%. The data implies that DCI (proposed) is the most efficient deduplication ratio and storage space savings method. The performance evaluation, which compared the proposed deduplication method with various state-of-the-art techniques, demonstrated that the new methodology has significant advantages over existing methods. The results regarding the deduplication ratio and the percentage of storage space saved confirm the efficiency of the proposed approach, showing that it outperforms current state-of-the-art methods. The performance improvement is attributed to the proposed method's consideration of both file-level and block-level approaches, facilitating optimal decision-making in the deduplication process.

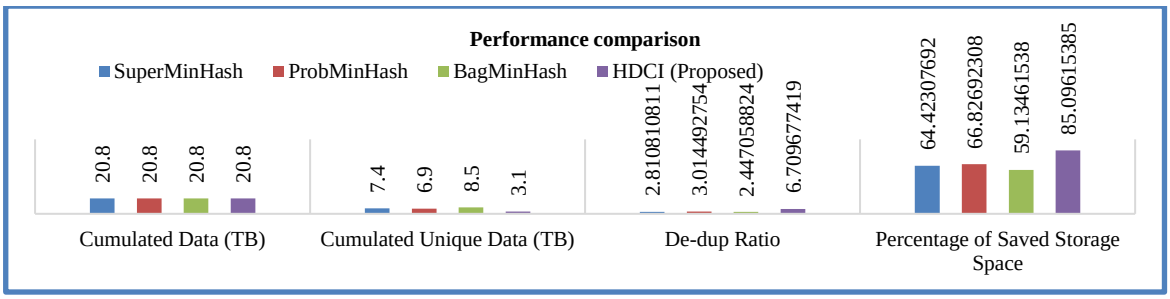

Figure 8. Performance comparison among deduplication methods

6. DISCUSSION

The proposed cloud deduplication methodology supports both file and block-level deduplication processes. This approach is more efficient because it considers file and block-level deduplication procedures, leading to higher efficiency in a cloud storage infrastructure. The proposed system has mechanisms to deal with distributed storage infrastructure and improve its efficiency with the deduplication procedure. Since cloud infrastructure stores and manages the data of billions of users across the globe, it is essential to have unique copies of data. In contrast, duplicate copies will hold a pointer to the original data, saving storage space and making the infrastructure energy efficient. The provision to manage metadata in the metadata server in the proposed system has the potential to improve accuracy in the identification of duplicate objects. The proposed algorithm is efficient with both file and block-level deduplication processes. The proposed deduplication methodology is specifically designed for cloud infrastructures. Given that these infrastructures handle large volumes of data, the deduplication process effectively eliminates duplicate multimedia objects, ensuring that only unique content is maintained. Instead of creating duplicate copies, the system allows for the reference of these objects when necessary, which helps conserve storage space and brings additional

benefits. This methodology has been shown to be more efficient than existing methods, as it operates effectively at both file and block levels, leading to enhanced optimization of the deduplication process. Furthermore, when applied to distributed storage facilities, the proposed methodology can significantly improve infrastructure efficiency and utility over time. The optimization achieved can lead to numerous advantages, such as increased customer satisfaction, better adherence to SLAs, enhanced energy efficiency in cloud data centers, and improved resource optimization. However, the proposed system has certain limitations, as expressed in section 6.1.

6.1. Limitations

The proposed deduplication methodology has certain limitations. First, the methodology is designed to deal with data DCI. Its scope is limited to textual data or data in different documents. It has no provision for supporting the deduplication of image objects in the cloud. Another significant limitation is that the proposed method exploited three different datasets to conclude. However, it is understood that there is a need to conduct empirical studies with more diversified datasets to draw general conclusions. Yet another significant limitation of the proposed methodology is that it needs to explore learning-based approaches that are important in the era of AI.

7. CONCLUSION

The primary objective of this research was to optimize cloud infrastructure efficiency through a novel data deduplication methodology. The proposed DCI algorithm combines file-level and block-level deduplication to address the challenges of redundant data storage. Experimental results validated the effectiveness of the algorithm, achieving a deduplication ratio as high as 6.7 and saving up to 85.09% of storage space. These significant gains highlight the capability of DCI to handle large-scale data efficiently and provide measurable improvements over existing methods. The implementation of the DCI algorithm has a direct and profound impact on cloud infrastructure performance. By eliminating redundant data and optimizing storage allocation, the DCI algorithm reduces the storage requirements significantly, ensuring better utilization of cloud resources. By reducing the volume of data stored and processed, the algorithm minimizes energy consumption in data centers, contributing to a more sustainable cloud computing environment. The incorporation of an optimal redundancy strategy ensures high availability and faster data access, thereby enhancing user satisfaction and adherence to SLAs. While the proposed methodology has shown promising results, it is important to acknowledge certain limitations. The current system is tailored to textual and document-based data, with no provision for image deduplication. Additionally, the experimental evaluation utilized three datasets, and further studies with diversified datasets are necessary to draw more generalized conclusions. Furthermore, the methodology does not currently leverage learning-based approaches, which could be pivotal in the era of AI. In future work, we aim to expand the scope of the DCI algorithm to include image deduplication and explore deep learning techniques for more adaptive and intelligent deduplication. These advancements will pave the way for the development of smarter, more efficient deduplication methods that address the dynamic needs of modern cloud infrastructures.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mohd Hasan Mohiuddin	✓	✓			✓	✓		✓	✓	✓	✓			✓
Latha Tamilselvan	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing – **O**riginal Draft
E : Writing – Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] R. Kaur, I. Chana, and J. Bhattacharya, "Data deduplication techniques for efficient cloud storage management: a systematic review," *Journal of Supercomputing*, vol. 74, no. 5, pp. 2035–2085, 2018, doi: 10.1007/s11227-017-2210-8.

[2] J. G. Jeslin and P. M. Kumar, "Decentralized and Privacy Sensitive Data De-Duplication Framework for Convenient Big Data Management in Cloud Backup Systems," *Symmetry*, vol. 14, no. 7, pp. 1–20, 2022, doi: 10.3390/sym14071392.

[3] V. Lytvyn, V. Vysotska, M. Osypov, O. Slyusarchuk, and Y. Slyusarchuk, "Development of intellectual system for data de-duplication and distribution in cloud storage," *Webology*, vol. 16, no. 2, pp. 1–42, 2019, doi: 10.14704/web/v16i2/a188.

[4] Z. Yan, L. Zhang, W. Ding, and Q. Zheng, "Heterogeneous data storage management with deduplication in cloud computing," *IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 393–407, 2019, doi: 10.1109/TBDATA.2017.2701352.

[5] B. Venkatesan and S. Chitra, "Data De-Duplication Process and Authentication Using ERCE with Poisson Filter in Cloud Data Storage," *Intelligent Automation and Soft Computing*, vol. 34, no. 3, pp. 1603–1615, 2022, doi: 10.32604/iasc.2022.026049.

[6] M. N. U. Khan, W. Cao, Z. Tang, A. Ullah, and W. Pan, "Energy-Efficient De-Duplication Mechanism for Healthcare Data Aggregation in IoT," *Future Internet*, vol. 16, no. 2, pp. 1–21, 2024, doi: 10.3390/fi16020066.

[7] G. Geetha, "Secure And Efficient Of Cloud Storage With Data De-Duplication," *International Journal of Innovative Research and Advanced Studies (IJIRAS)*, vol. 5, no. 4, pp. 1–11, 2018.

[8] Y. Gao, K. Li, and Y. Jin, "Compact, popularity-aware and adaptive hybrid data placement schemes for heterogeneous cloud storage," *IEEE Access*, vol. 5, pp. 1306–1318, 2017, doi: 10.1109/ACCESS.2017.2668392.

[9] G. Said *et al.*, "Hash Table Assisted Efficient File Level De-Duplication Scheme in SD-IoV Assisted Sensing Devices," *Intelligent Automation and Soft Computing*, vol. 38, no. 1, pp. 83–99, 2024, doi: 10.32604/iasc.2023.036079.

[10] C. Esposito, "Interoperable, dynamic and privacy-preserving access control for cloud data storage when integrating heterogeneous organizations," *Journal of Network and Computer Applications*, vol. 108, pp. 124–136, 2018, doi: 10.1016/j.jnca.2018.01.017.

[11] S. Bose, D. Sarkar, and N. Mukherjee, "A Framework for Heterogeneous Resource Allocation in Sensor-Cloud Environment," *Wireless Personal Communications*, vol. 108, no. 1, pp. 19–36, 2019, doi: 10.1007/s11277-019-06383-1.

[12] M. Sohani and S. C. Jain, "A Predictive Priority-Based Dynamic Resource Provisioning Scheme with Load Balancing in Heterogeneous Cloud Computing," *IEEE Access*, vol. 9, pp. 62653–62664, 2021, doi: 10.1109/ACCESS.2021.3074833.

[13] X. Ma, W. Yang, Y. Zhu, and Z. Bai, "A Secure and Efficient Data Deduplication Scheme with Dynamic Ownership Management in Cloud Computing," in *Conference Proceedings of the IEEE International Performance, Computing, and Communications Conference*, 2022, pp. 194–201, doi: 10.1109/IPCCC55026.2022.9894331.

[14] E. Pugazhendi, M. R. Sumalatha, and P. L. Harika, "Weight based deduplication for minimizing data replication in public cloud storage," *Journal of Scientific and Industrial Research*, vol. 80, no. 3, pp. 260–269, 2021, doi: 10.56042/jsir.v80i03.41337.

[15] E. Mohan, R. Anandan, and S. Shakthibalan, "The Data De-Duplication In Cloud Storage Management," *Nat. Volatiles & Essent. Oils*, vol. 8, no. 2, pp. 118–124, 2021.

[16] K. Vijayalakshmi and V. Jayalakshmi, "Analysis on data deduplication techniques of storage of big data in cloud," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, 2021, pp. 976–983, doi: 10.1109/ICCMC51019.2021.9418445.

[17] F. Gang and D. Wei, "Dynamic Deduplication Algorithm for Cross-User Duplicate Data in Hybrid Cloud Storage," *Security and Communication Networks*, pp. 1–10, 2022, doi: 10.1155/2022/8354903.

[18] A. H. Adhab and N. A. Hussien, "Techniques of Data Deduplication for Cloud Storage: A Review," *International Journal of Engineering Research and Advanced Technology*, vol. 08, no. 04, pp. 07–18, 2022, doi: 10.31695/ijerat.2022.8.4.2.

[19] P. Prajapati and P. Shah, "A Review on Secure Data Deduplication: Cloud Storage Security Issue," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 3996–4007, 2022, doi: 10.1016/j.jksuci.2020.10.021.

[20] G. Sujatha and J. R. Raj, "A Comprehensive Study of Different Types of Deduplication Technique in Various Dimensions," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, pp. 316–323, 2022, doi: 10.14569/IJACSA.2022.0130339.

[21] T. K. Selvi and S. Sasirakha, "Data Management Issues and Study on Heterogeneous Data Storage in the Internet of Things," *Computer Science & Engineering: An International Journal*, vol. 12, no. 6, pp. 27–34, 2022, doi: 10.5121/cseij.2022.12604.

[22] G. Neelamegam and P. Marikkannu, "Health Data Deduplication Using Window Chunking-Signature Encryption in Cloud," *Intelligent Automation and Soft Computing*, vol. 36, no. 1, pp. 1079–1093, 2023, doi: 10.32604/iasc.2023.031283.

[23] J. Kim, S. Ryu, and N. Park, "Privacy-enhanced data deduplication computational intelligence technique for secure healthcare applications," *Computers, Materials and Continua*, vol. 70, no. 2, pp. 4169–4184, 2022, doi: 10.32604/cmc.2022.019277.

[24] S. Borade, A. Khan, A. Khan, A. Sayyed, and Ranjan, "Image and Text Encrypted Data with Authorized Deduplication in Cloud," *International Research Journal of Innovations in Engineering and Technology (IRJIET)*, vol. 7, no. 5, pp. 278–282, 2023.

[25] R. Arora and D. Vetrithangam, "Advancements in Deduplication Techniques for Efficient Data Storage," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 5, pp. 2128–2151, 2024.

[26] N. R. H. Manikyam and M. S. Devi, "An Image Decompression Model with Reversible Pixel Interchange Decryption Model Using Data Deduplication," *Traitement du Signal*, vol. 39, no. 1, pp. 195–203, 2022, doi: 10.18280/ts.390120.

[27] M. Nagappan, J. Swapna, A. Pandiaraj, R. Rajakumar, and K. Moez, "Hybrid cloud storage system with enhanced multilayer cryptosystem for secure deduplication in cloud," *International Journal of Intelligent Networks*, vol. 4, pp. 301–309, 2023, doi: 10.1016/j.ijin.2023.11.001.

[28] E. Manogara, and S. Abirami, "A smart hybrid content defined chunking algorithm for data deduplication in cloud storage," *Research Square*, pp. 1-29, 2022, doi: 10.21203/rs.3.rs-376128/v1.

[29] O. Ertl, "ProbMinHash – A Class of Locality-Sensitive Hash Algorithms for the (Probability) Jaccard Similarity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3491–3506, 2020, doi: 10.1109/TKDE.2020.3021176.

[30] O. Ertl, "SuperMinHash - A New Minwise Hashing Algorithm for Jaccard Similarity Estimation," *arXiv*, pp. 1–6, 2017, doi: 10.48550/arXiv.1706.05698.
[31] O. Ertl, "BagMinHash - minwise hashing algorithm for weighted sets," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Jul. 2018, pp. 1368–1377, doi: 10.1145/3219819.3220089.

BIOGRAPHIES OF AUTHORS

Mohd Hasan Mohiuddin is a dedicated researcher and Ph.D. student in the Department of Computer Science and Engineering at B.S. Abdur Rahman Crescent Institute of Science & Technology, Vandalur, India. He received his master's degree in Computer Science and Engineering at VIF College of Engineering, Affiliated to Jawaharlal Nehru Technological University Hyderabad, Telangana, India. His research interests are cloud computing, cyber security. artificial intelligence, machine learning applications, and optimization methods. He can be contacted at email: Mohiddin.hasan@outlook.com.

Dr. Latha Tamilselvan is working as a professor in the department of Information Technology, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. She received her Ph.D. from Anna University, Chennai, India. She has more than 25 years of Academic Experience. Her research interests include mobile ad hoc networks, network security, cloud computing, and IoT. She is working as reviewer for noteworthy journals that are Scopus, SCI indexed journal. She has published around 50 research papers, in international journals, International Conferences and National conferences. She can be contacted at email: latha.tamil@crescent.education.