Efficient and Secure Data Deduplication in Distributed Cloud Storage: A Fault-Tolerant Model Using Active Learning for Big Data Management

S. Seethalakshmi[1], Dr. B. Balakumar[2]

[1]Research Scholar, Manonmaniam Sundaranar University, Tirunelveli. Tamilnadu, India
[2]Assistant Professor, Manonmaniam Sundaranar University, Tirunelveli. Tamilnadu, India

ABSTRACT

As the big data era advances, more and more redundant data are being expressed in various ways. Data deduplication knowledge has never been more important than it is now for lowering redundant data storage and enhancing data quality. Connecting several data tables and identifying distinct entries that point to the same item is typically required, particularly when multi-source data deduplication is involved. Active learning minimises the amount of data that needs to be annotated and trains the classical by choosing the data pieces with the greatest evidence divergence. This approach offers special benefits when handling massive data annotations. Unfortunately, the majority of existing active learning techniques are rarely used for data deduplication jobs and only use classical entity matching. The research effort suggests a model, a distributed cloud storage method that ensures effectiveness, security, and high availability, to close this research gap. By intelligently distributing data subsets among servers, the suggested approach achieves fault tolerance while preserving redundancy for high availability. This suggested model reduces the amount of storage space and upkeep required for the data while eliminating duplicate copies. Additionally, the data is kept in a highly secure and effective manner. The suggested model's efficacy in fault tolerance, cost reduction, batch auditing, and block- and file-level deduplication is demonstrated by the experimental findings. It performs better than current systems thanks to its robust fault tolerance, minimal time complexity, and excellent deduplication capabilities.

Keywords: Duplicate Data; Fault Tolerance; Multi-source data; Storage Space; Cloud Computing.

INTRODUCTION

Large volumes of data must be stored by both individuals and businesses in the age of the data explosion. For instance, according to IDC, by 2025, the world's datasphere would have grown to 175 zettabytes [1]. Users would want to transfer their data to a server in order to alleviate the effort of managing and storing massive amounts of data when faced with a significant storage demand. However, because several identical pieces of data may be uploaded to the cloud server by many users, it results in a large amount of duplicate data being stored there [2]. According to the study, roughly 75% of data in conventional application systems is duplicated, and in backup and archival storage systems, the percentage of identical data even exceeds 90% [3]. The advent of deduplication technology allows cloud servers to store less redundant data. Thanks to deduplication technology, several identical pieces of data are stored on the cloud server in only one duplicate [4]. Due to its ability to save significant costs for both the customer and the cloud server, this technology is receiving a lot of attention.

Applications for data deduplication are numerous and span several industries, including cloud-based electronic health systems [5]. Cloud- systems are more effective, precise, and dependable at records than traditional medical record management systems [6]. Furthermore, the utilisation of cloud-assisted electronic health systems is crucial in the settlement of medical malpractice verdicts and disputes. As everyone is aware, there is a limited amount of diagnostic data, including medication interactions and symptoms, in electronic medical records. As an illustration, there are currently only roughly 100 different types of antibiotics [7]. As a result, electronic medical records include a great deal of identical information. According to the study, data deduplication in electronic health systems can save over 65% of storage space [8].

In order to do study on specific disorders, medical researchers need to share data from electronic medical records. For instance, common symptoms of the patient, such as fever and dizziness, are recorded in the COVID-19 electronic medical record [9]. Exchanging electronic health records is a great way for researchers to learn where to look for possible COVID-19 cases. However, there may be privacy-exposure issues if electronic medical records are shared. The electronic medical record typically consists of two sections [10]. Sensitive information, including the patient's, is contained in the first section. The doctor's suggested diagnostic data, which includes the patient's symptoms, the nature of the ailment, the dosage of the drug, and other details, is included in the second section. In reality, researchers should only use diagnostic data; they shouldn't obtain sensitive patient information through data sharing services [11]. Thus, achieving data sharing is crucial, provided that data deduplication is done well and that the private information contained in kept hidden.

Traditional encryption schemes, which are often designed using public key infrastructure, struggle with significant processing overheads and large network bandwidth utilisation as the need for complex access control policies and data sharing continues to rise [12]. Attribute-Based Encryption (ABE) techniques have attracted a lot of attention recently as a possible remedy for these issues [13]. Reducing duplicate content in cloud storage lowers the cost of storage. Since the cloud server is managed by a third party and is therefore frequently viewed as unreliable, all data stored there is encrypted before being transferred, which affects de-duplication because of encryption's unpredictability property [14]. The cloud server may limit access to sourced information even in cases where ownership is subject to dynamic ownership changes [15].

The impetus for this study is the growing significance of safe and effective data handling in the context of cloud computing. The increasing dependence of both individuals and organisations on cloud service providers for data transmission and storage has made it critical to guarantee privacy and efficient use of resources [16]. Even if they work well, traditional data deduplication techniques run into problems when combined with the need for encryption to increase security. A novel approach was developed in response to this contradiction between data encryption and deduplication efficiency, and this approach serves as the foundation for the suggested paradigm. The fundamental idea behind this model is the combination of a biassed sampling-based bloom filter with a hybrid chunking technique that uses a window size chunking algorithm. In the Fog-Cloud situation, it offers encrypted data via the ASE algorithm. The suggested work's primary contributions are:

- The window size chunking algorithm is being implemented to divide the data stream.
- To remove redundant data by employing the Bloom filter.
- To employ an innovative signature-based encryption technique to safeguard and maintain privacy of data stored cloud.

Section 2 of the study outlines the literature review, Section 3 delineates the system model, and Section 4 offerings the deduplication utilising the proposed model. Section 5 deliberates on the experimental outcomes, and Section 6 culminates the work with recommendations for future research.

RELATED WORKS

In order to extract features of records, Cao et al. [17] have proposed a novel graph deep deduplication. This framework first introduces the graph active learning strategy in order to build a clean graph that is used to filter the labelled, which is used to delete the duplicate data that retains the most information. The suggested approach performs better on data deduplication tasks than the most advanced active learning models, according to experimental results on real-world datasets.

A hybrid cloud-based secure deduplication strategy designed for large-scale data system deployment has been introduced by Tang et al. [18]. Our method specifically makes use of ciphertext-policy attribute-based encryption (CP-ABE), which allows us to set up key management and access control through a private cloud server. In addition, to use a public cloud server to support businesses and organisations looking for safe storage for their data. Notably, our method uses ElGamal encryption to offer mutual zero-interaction verification between public and private cloud servers, ensuring data unforgeability. The security evaluation demonstrates how our suggested method protects the integrity and privacy of data. to also demonstrate how the method achieves secure and effective access control and key management, thwarting hostile users' attempts to trick cloud servers into returning wrong ciphertext using brute-force assaults on the dictionary. Additionally, performance and functional examination highlights our method's advantages over five other traditional data deduplication approaches. The scheme's performance remains high throughout, even with the assumption of having more extensive security settings.

Guo, Xian, and colleagues [19] have introduced an encrypted data deduplication system called AF-Dedup and proposed a unique data structure for PoW called adaptive dynamic Merkle hash forest (ADMHF). It lessens the chances of data content exposure brought on by repeated attempts at ownership verification in conventional systems. In particular, to create the file tag first in order to serve as the file's unique identification. Second, based on how well-liked the data is, various encryption techniques are used. After that, the matching ADMHF is produced in order to facilitate further ownership confirmations. It has been demonstrated through security research and simulation trials that our technique greatly improves small file security. Our technique achieves the same level of security as the current scheme for a file with 91 blocks in a given scenario for files with only two blocks.

To improve deduplication performance in cloud storage, novel hybrid chunking techniques have been proposed by Akbar et al. [20]. To reduce the deduplication data, a content-based hybrid chunking approach combining a Dynamic Prime Coding (DPC) algorithm with a Two Threshold Two Divisor (TTTD) algorithm is recommended. An index approach needs to be used in the deduplication matching process in order to boost throughput, which translates to a reduction in utilisation. It provides more security protection for chunks security levels and compromises security for greater deduplication effectiveness for chunks with lower security levels. Identity management, safe data interchange, and privacy preservation are more difficult privacy are easier to handle. In light of this, the paper suggested a cross-domain authentication method for the validation procedure. The effectiveness of the concept is demonstrated by assessments of real-world datasets. Several performance characteristics, such as chunk distribution, processing size, and deduplication ratio, are analysed using the suggested hybrid Content-based TTTD-DPC using different techniques. The suggested method improves both the overall throughput and processing speed of the deduplication scheme. Furthermore, in comparison to other current approaches, the suggested methodology lowers the computational cost (772 ms) hash function.

A quantum safe hybrid level deduplication technique, has been suggested by Khan et al. [21]. It offers protection against post-quantum attacks. Our underlying user data deduplication system with inbuilt Proof of Ownership (PoW) and Proof of Storage (PoS) security services, in contrast to CE and MLE based deduplication solutions. Higher levels of security in the post-quantum era are provided by the security analysis of the proposed HLSBD2 scheme. Furthermore, the scheme's performance analysis shows how effective it would be if implemented in real life.

For data storage in cloud computing, Pavithra et al., [22] have proposed Boneh Goh Nissim Bilinear control and safe de-duplication. The suggested approach achieves reduced compute consumption while achieving fine-grained access control. to create Boneh Goh Nissim Privacy Preserving Revocable Attribute-based Encryption, which prevents the release of sensitive data and strengthens attribute revocation. Additionally, to use rand pattern matching to conceal the access patterns by preventing publication of the patterns through the use of the Optimal Cache Oblivious method. In order to reduce the duplication and encryption operations simultaneously, to support updating policies. To increase data confidentiality and integrity, to share data securely. Ultimately, our comprehensive cloud experiments showed that our suggested BGNBA-OCO approach is more effective than previous studies.

Problem statement

Securing the data deduplication model for the cloudfog storage integrated environment is the main objective of this suggested study. Numerous studies on data deduplication have been conducted. Nevertheless, the shortcomings of the current, conventional algorithms include their poor ability to identify data redundancy and storage space.

SYSTEM MODEL AND DESIGN PRINCIPLES

The functionality of the proposed system and a number of data structures are defined in this part first. These will be used later on by several user-level processes to carry out read, write, update, besides delete activities. Here is a high-level summary of the entire procedure.

A. Index Server Structure and Initialization

The Index server also maintains a set of records pertaining to users, data servers, and previously submitted material. The following are the data structures that are stored:.

Cloud Users. Users (U) is an unordered map construction with unique user IDs (U_{id}) containing multiple files (F). Each file has lists (H_{k}). Any number of U_{id}, F_{id}, or H_{k} can be inserted, replaced, or deleted in continuous time complexity.

Data Server. The data server (S_{i}) are documented by unique keys (S_{id}). In each Sid, a boolean list is created (i.e., TRUE: not available; FALSE: space obtainable) with the corresponding storage site A as key.

HashMap. Our An effective and straightforward unordered map structure, HashMap (HM) uses a hash function. value (H_{k}) as the key, where each H_{k} is a general-purpose tag for data block (B_{i}). For each Hk, several pairs of ($S_{\mathrm{id}_{\mathrm{j}}}$, A_{j}) can be used to provide the precise location of memory and the server name for data archiving and retrieval.

MHT Construct. to use our structure to build a Merkel Hash Tree (MHT) to confirm data integrity. First, the Merkle tree approach is used to generate a Merkle root Mr from any file F. The user verify the verifiability of individual data blocks once all file blocks have been uploaded to cloud servers. The server answers the challenge query in response. To ensure data integrity, the user uses the server's answer to recalculate the Merkle root and sees if it matches the stored root.

B. Data Processing

The following techniques are employed by our system to process data: Files are split into data blocks based on a given block size, ii) AES-256 symmetric file, iii) encryption generates keys by hashing each file, and iv) the SHA-256 algorithm provides authentication tags for each block.

PROPOSED METHODOLOGY

The decloud-fog-based data is reduced and safely stored by this proposed method. There are two phases to it.;

Phase 1:keeps the data as a single copy and removes duplicates size chunking approach.

Phase 2:utilising the ASE technique to securely store the data cloud-fog-based situation.
the information that is gathered from multiple sources and sent to data consumers. Next, the redundant data is removed by the deduplication process, which combines a biassed sampling-based bloom filter with a window size chunking method. It then goes to the FoG server following the elimination procedure. The ASE algorithm is used by the FoG server to upload the safe data after processing it in accordance with the cloud format.

De-Duplication

The suggested work implements a window technique with a biassed sampling-based bloom filter to remove the de-duplicated data. There are two steps involved in implementing this de-duplication technique.;

Stage 1:using the Window Size Chunking Algorithm to split data according to window size (WCA).

Stage 2: The window size value using a biassed sampling-based Bloom is computed based on the input.

Window Size Chunking Algorithm (WCA)

It regulates the chunk size according to the window's size threshold worth. The management of the data chunks, data deduplication, and cloud storage are the main responsibilities of the FoG server. ":" is used to separate data that is read in string format from multiple devices. The entire piece of information is divided for each user by ";". It alludes to the user as a singular entity. The Window Size is demonstrated in Algorithm 1.

Algorithm 1: Window-Size Based Chunking Procedure

```
$Input$: $Collected\ data\ in\ a\ string\ format\ str\ hcd$
$Output$: $Chunk\ of\ string\ data\ in\ a\ window\ size\ wind$
$Predefined\ Value$: $Delimiter\ (;)$
$Step\ 1$: $Spilt\_str = split(str\_hcd;\ D)$
$Step\ 2$: $for\ i = 0\ to\ Split\_str: size$
$Step\ 3$: $if\ (selected\_str: size\ >\ wind)$
$Step\ 4$: $selected\_str.remove(selected\ str: size\ \ -1)$
$Step\ 5$: $Return\ selected\ str$
$Step\ 6$: $Break$
$Step\ 7$: $Else\ If\ (selected\ str: size\ ==\ wind)$
$Step\ 8$: $Return\ selected\_str$
$Step\ 9$: $Break$
$Step\ 10$: $Else\ If\ (selected\_str: size\ > 3/4\, wind)\ \&\&\ (selected\_str: size\ <\ wind)$
$Step\ 11$: $Return\ selected\_str$
$Step\ 12$: $Break$
$Step\ 13$: $Else$
$Step\ 14$: $selected\_str[i] = split\_str[i]$
$Step\ 15$: $End\ If$
$Step\ 16$: $End\ For$
```

The data is taken as input in Algorithm 1, separated by a separator (;) between each record, and then saved in an type called split_str[i] for value. All data elements are traversed using the FOR loop. It indicates that the data is saved in another array if it doesn't go beyond the window size of more than 75% of the data. Three criteria determine the window's chunk size: if the window size (wind), the deleted and the array is returned. Second, it returns the array value exactly if array size equals window size (wind). The chosen data in the range among ¾ and its window size is then used to determine the array size. It provides the range of chunks that was chosen.

Biased Sampling-Based Bloom Filter

This method is applied to the result of Algorithm 1 to de-duplicate a large amount of stream data. Bloom filters B are arrays with n by b bits in size. It is initially given the number 0. The addition of a new filter and the data element it belongs to is m∈M. In uniform hash functions which are arbitrarily selected. The arbitrarily selected p self-governing hash functions are defined as $hash_h(.)$ and 1≤h≤p.

- ❖ At first, the hash function $hash_h(de)$ to element d is estimated.
- ❖ To get position pos in the array n using modulo function md, it is applied to all elements. Then the bloom filter is indistinct as $bf_h(de) = hash_h(de)Þmodmd$, where $bf_h(x) \in [0, md - 1]$.
- ❖ To add a new data element, d, and determine if it is set or not by looking at the position of its b bits. The insertion operation is deemed complete when all b bit position elements are set to 1 following the insertion of the data element.The data element de is placed at 0 and removed from the bloom filter.
- ❖ The bit situation of the current element is examined; if it contains the value 1, the bloom filter views the element as distinct; if not, it deems it to be duplicate.

The size of filter and the stream's current length are shown here. The number of bits in the bloom filter rises to the set value, which also causes the distinct element to be incorrectly reported as duplicate. If both the likelihood of the same element rate and the duration of the healthcare data stream rise. All bits are set to 1 when elements from the health care data stream are inserted into the bloom filter. That rate of false positives is that. In order to get around this problem, whenever a new element is added to the bloom filter array, a component is randomly deleted and its value is set to 0. It treats its duplicate value as separate and produces false negatives.

False Positive Rate (FPR) occurs when a unique data element is reported as duplicate. However, duplicate stream data is recorded as separate if False Negative Rate (FNR) is seen.

Let $de_{\mathrm{m}+1}$ and (m=1)$^{\text{th}}$; data part of smearing hash function. If it meets the criteria in Step 6, the bloom filter report is deemed a duplicate. Otherwise, it is regarded distinct.

Encrypted Data Storage

This suggested Advanced Signature- strategy is used to manner. The privacy of user data is safeguarded by the ASE system. It restricts access to the user's data to only those who are authorised. The user's data can returned to the specific patient by using the ASE signature technique.

E-data has composed all the data from diverse sources. Then each data is alienated into $(hc_1, hc_2, hc_3, \ldots, hc_n)$ along with user's pair of $(prik_i, sig_i)$. To store the data from fog, $user_1$ using key $prik_1$ and sign key sig_1 and gets the signature sign $(user_1)$ using ASE arrangement. Similarly, $(user_2)$ using their private key $prik_2$besides sign key sig_2 will sign $(user_1, user_2)$ scheme and so on. In get the key $pubk$ of user authority (UA) is added. Only then, the data storage.

The sequential aggregate signature scheme is used to generate the digital signature for the ASE with validating signature creation. Less storage space and greater security are provided. The advanced signature-based algorithm concept is put into practice.

Verification of Digital Signature

It uses "ID" to identify different types of sensor devices. Sensor device private keys that generate random facts are used to issue the certificate of authentication. The server does the mapping operation to identify the sensor device for the specific patient.

Data deduplication

Since to can assume that every connected node in a clean points to the same thing, the redundant data needs to be removed. All of the data in a network must have similarity scores intended, score must be retained. The computation for the reserved data R is given by equation (1), where k is the sum of nodes in a graph and w is the similarity score between this node and another node.

$$R = \begin{cases} \forall w_k, \ \ k = 2 \\ max \sum_{k=1}^{n-1} w_k, k > 2 \end{cases} \quad (1)$$

Among all nodes pointing to the same entity, the data with the highest sum of similarity scores—calculated using Equation (1)—can be regarded as the most representative data of that entity; other data should be destroyed, while these data should be kept. If there are just two pieces of data, choose the ones with less missing values to keep. The deletion of any data is deemed possible if there are no missing values.

RESULT AND ANALYSIS

This section includes a brief description of the experimental setup, results analysis, and dataset used in this work.

Dataset

The MusicBrainz dataset is one of four datasets used in the study's evaluation of the suggested model [23]. This well-known dataset is utilised, particularly in the field of music, for multi-source entity recognition. It includes an extensive collection of entities linked to music, including songs, albums, artists, and the metadata that goes along with them. The dataset is extensively utilised in the study and advancement of entity recognition and disambiguation systems in the music domain. The Magellan repository is the source of the second dataset [24]. It includes details on various restaurants, such as their names, locations, phone numbers, menus, reviews, and other characteristics. The dataset is frequently used to test and compare different entity-matching methods and algorithms. This dataset is used by practitioners and researchers to test and develop methods for finding and fixing duplicate or matched restaurant entries across numerous sources. The latest one [25] is based on a subset of computer product records that have been made available online by four e-commerce sites and one of its more complicated and sparser variations. The goal of the dataset is to replicate the difficulties involved in finding and removing duplicate product records from various web sources. These datasets consist of numerous tables with identical features from various data sources. Specific dataset information is given in Table 1. The ratio of missing attribute values is known as sparsity.

Table 1: Details of four datasets in experiments.

Multi-source dataset	Number of data sources	Number of matched pairs in1000	Number of non-matched pairs in1000	Range of sparsity
Computer	4	4.8	69.6	[0−0.05]
Computer_mut	4	4.8	69.6	[0−0.18]
Restaurants	4	11.2	56.5	[0−0.08]
MusicBrainz	5	16.1	369.7	[0.05−0.12]

Validation analysis of the anticipated model

Table 2 and 3 provides the experimental analysis of the projectedperfect with existing techniques

Table 2: Execution period for beforehand smearing de-duplication data

Execution time in (ms)	De-duplication file size (MB)				
Technique name	**200**	**400**	**600**	**800**	**1000**
AES	1200	3600	5000	7500	9300
DES	1350	3780	5400	7800	9800
Blowfish	1175	3450	4800	6400	8900
Twofish	1190	3400	4725	6325	8650
Proposed	975	3100	4600	5900	8400

The execution period for different techniques applied before de-duplication shows that the proposed method continues to outperform AES, DES, Blowfish, and Twofish in terms of speed. For a 200 MB file, the proposed method's execution time is 975 ms, which is faster than AES (1200 ms), DES (1350 ms), Blowfish (1175 ms), and Twofish (1190 ms).As the file size increases to 400 MB, the proposed method takes 3100 ms, which is significantly faster than AES (3600 ms), DES (3780 ms), Blowfish (3450 ms), and Twofish (3400 ms). For 600 MB, the proposed method's execution time is 4600 ms, while AES, DES, Blowfish, and Twofish take 5000 ms, 5400 ms, 4800 ms, and 4725 ms, respectively. Lastly, for a 1000 MB file, the proposed method completes the execution in 8400 ms, compared to AES (9300 ms), DES (9800 ms), Blowfish (8900 ms), and Twofish (8650 ms).In conclusion, the proposed method consistently provides faster execution times before de-duplication, showing a clear advantage over the other techniques for all file sizes.

Table 3: Execution period for afterward smearing de-duplication statistics

Execution time in (ms)	**After de-duplication file size (MB)**				
Technique name	**200**	**400**	**600**	**800**	**1000**
AES	1100	3550	4970	7370	9280
DES	1320	3720	5370	7750	9750
Blowfish	1150	3370	4650	6350	8760
Twofish	1160	3300	4625	6225	8450
Proposed	875	2900	4450	5700	8320

The execution period for different techniques after applying de-duplication statistics shows that the proposed method consistently outperforms AES, DES, Blowfish, and Twofish in terms of efficiency. For a 200 MB file, the proposed method has an execution time of 875 ms, compared to 1100 ms for AES, 1320 ms for DES, 1150 ms for Blowfish, and 1160 ms for Twofish. As the file size increases to 400 MB, the proposed method takes 2900 ms, which is faster than AES (3550 ms), DES (3720 ms), Blowfish (3370 ms), and Twofish (3300 ms). For 600 MB, the proposed method's execution time is 4450 ms, while AES, DES, Blowfish, and Twofish take 4970 ms, 5370 ms, 4650 ms, and 4625 ms, respectively. At 800 MB, the proposed method completes in 5700 ms, compared to AES (7370 ms), DES (7750 ms), Blowfish (6350 ms), and Twofish (6225 ms). Finally, for a 1000 MB file, the proposed method has an execution time of 8320 ms, outperforming AES (9280 ms), DES (9750 ms), Blowfish (8760 ms), and Twofish (8450 ms).The results clearly demonstrate that the proposed method provides a more efficient execution period across all file sizes compared to the other techniques.

CONCLUSION

This study introduced a deep active deduplication that is based on similarity algorithms in conjunction with an encryption model to extract features of data records. Additionally, the active learning policy was introduced to filter the data that requires labelling, which is rummage-sale to remove duplicate data that efficiently retains the most info. Based on four multi-source tasks, the experimental results demonstrate that the suggested model outperforms baseline methods in characterising the attributes of various data records. In addition, the enhanced active learning outperforms both the baseline model and the most advanced committee-based query technique. The actual results show that our new approach works especially well on medium-sized datasets, and performance gains are shown when the matching ratio rises. Our approach performs exceptionally well when compared to other balance among computational time efficiency and experimental results. Subsequent developments will likely focus on improving the graph's filtration phase. For graph generation and data cleaning, a graph neural network is utilised to gather data from nearby nodes and progressively integrate both local and global information.

REFERENCES

[1]. Rajagopal, M., Ramkumar, S., & Ganesh, L. (2023, June). Probabilistic Data Structure Using Hashing Technique for Big Data Security De-duplication in Cloud Environment. In International Conference on Data Science and Big Data Analysis (pp. 125-134). Singapore: Springer Nature Singapore.

[2]. Baligodugula, V. V., Amsaad, F., Tadepalli, V. V., Radhika, V., Sanjana, Y., Shiva, S., ... &Tashtoush, Y. (2023, May). A Comparative Study of Secure and Efficient Data Duplication Mechanisms for Cloud-Based IoT Applications. In International Conference on Advances in Computing Research (pp. 569-586). Cham: Springer Nature Switzerland.

[3]. Devi, V. A., Thirumalraj, A., Kavin, B. P., & Seng, G. H. Securing the Predicted Disease Data using Transfer Learning in Cloud-Based Healthcare 5.0. In Intelligent Systems and Industrial Internet of Things for Sustainable Development (pp. 101-117). Chapman and Hall/CRC.

[4]. Mageshkumar, N., & Lakshmanan, L. (2023). Intelligent data deduplication with deep transfer learning enabled classification model for cloud-based healthcare system. Expert Systems with Applications, 215, 119257.

[5]. Rasina Begum, B., & Chitra, P. (2023). SEEDDUP: a three-tier SEcurE data DedUPlication architecture-based storage and retrieval for cross-domains over cloud. IETE Journal of Research, 69(4), 2224-2241.

[6]. Krishnasamy, V., & Venkatachalam, S. (2023). An efficient data flow material model based cloud authentication data security and reduce a cloud storage cost using Index-level Boundary Pattern Convergent Encryption algorithm. Materials Today: Proceedings, 81, 931-936.

[7]. Thirumalraj, A., Chandrashekar, R., Gunapriya, B., &kavin Balasubramanian, P. (2024). NMRA-Facilitated Optimized Deep Learning Framework: A Case Study on IoT-Enabled Waste Management in Smart Cities. In

Developments Towards Next Generation Intelligent Systems for Sustainable Development (pp. 247-268). IGI Global.

[8]. Peng, L., Yan, Z., Liang, X., & Yu, X. (2023). SecDedup: Secure data deduplication with dynamic auditing in the cloud. Information Sciences, 644, 119279.

[9]. Song, M., Hua, Z., Zheng, Y., Xiang, T., & Jia, X. (2023). FCDedup: A two-level deduplication system for encrypted data in fog computing. IEEE Transactions on Parallel and Distributed Systems.

[10]. Gunapriya, B., Thirumalraj, A., Anusuya, V. S., Kavin, B. P., & Seng, G. H. (2024). A Smart Innovative Pre-Trained Model-Based QDM for Weed Detection in Soybean Fields. In Advanced Intelligence Systems and Innovation in Entrepreneurship (pp. 262-285). IGI Global.

[11]. Neelamegam, G., &Marikkannu, P. (2023). Health Data Deduplication Using Window Chunking-Signature Encryption in Cloud. Intelligent Automation & Soft Computing, 36(1).

[12]. Ellappan, M., & Murugappan, A. (2023). A smart hybrid content-defined chunking algorithm for data deduplication in cloud storage. Soft Computing, 1-16.

[13]. Uma Maheswari, V., Stephe, S., Aluvalu, R., Thirumalraj, A., & Mohanty, S. N. (2024). Chaotic Satin Bowerbird Optimizer Based Advanced AI Techniques for Detection of COVID-19 Diseases from CT Scans Images. New Generation Computing, 1-23.

[14]. Keskin, S., &Isık, A. H. (2023). Examining The Importance of Artificial Intelligence In The Singularization Of Big Data With The Development Of Cloud Computing. International Journal of Engineering and Innovative Research, 5(2), 170-180.

[15]. Arunadevi Thirumalraj, A. K., & El-Sayed, M. (2024). ScatterNet-based IPOA for predicting violent individuals using real-time drone surveillance system. Industry 6.0: Technology, Practices, Challenges, and Applications, 182.

[16]. Wang, Z., Gao, W., Yang, M., & Hao, R. (2023). Enabling Secure Data sharing with data deduplication and sensitive information hiding in cloud-assisted Electronic Medical Systems. Cluster computing, 26(6), 3839-3854.

[17]. Cao, H., Du, S., Hu, J., Yang, Y., Horng, S. J., & Li, T. (2024). Graph Deep Active Learning Framework for Data Deduplication. Big Data Mining and Analytics, 7(3), 753-764.

[18]. Tang, X., Guo, C., Choo, K. K. R., Jiang, X., & Liu, Y. (2024). A secure and lightweight cloud data deduplication scheme with efficient access control and key management. Computer Communications, 222, 209-219.

[19]. Guo, X., & Xian, H. (2024). AF-Dedup: Secure Encrypted Data Deduplication Based on Adaptive Dynamic Merkle Hash Forest PoW for Cloud Storage. IEEE Transactions on Industrial Informatics.

[20]. Akbar, M., Ahmad, I., Mirza, M., Ali, M., &Barmavatu, P. (2024). Enhanced authentication for de-duplication of big data on cloud storage system using machine learning approach. Cluster Computing, 27(3), 3683-3702.

[21]. Khan, W. A., Khan, F., Tahir, S., Zhang, Y., Amjad, F., & Ahmad, J. (2024). HLSBD2: a quantum secure hybrid level source based data deduplication for the cloud. Journal of Ambient Intelligence and Humanized Computing, 15(1), 89-102.

[22]. Pavithra, M., Prakash, M., & Vennila, V. (2024). BGNBA-OCO based privacy preserving attribute-based access control with data duplication for secure storage in cloud. Journal of Cloud Computing, 13(1), 8.

[23]. A. Saeedi, E. Peukert, and E. Rahm, Using link features for entity clustering in knowledge graphs, in Proc. 15th Int. Conf., ESWC 2018, Heraklion, Greece, 2018, pp. 576–592.

[24]. P. Konda, S. Das, A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. Naughton, S. Prasad, et al., Magellan: Toward building entity matching management systems over data science stacks, Proceedings of the VLDB Endowment, vol. 9, no. 13, pp. 1581–1584, 2016.

[25]. A. Primpeli, R. Peeters, and C. Bizer, The WDC training dataset and gold standard for large-scale product matching, in Proc. 2019 World Wide Web Conf., San Francisco, CA, USA, 2019, pp. 381–386.