International Journal of
Computational Learning & Intelligence
Vol. 04 | Issue 04 | 2025

RESEARCH ARTICLE OPEN ACCESS

Secure Approach to Textual Data Deduplication in Cloud Systems: A Process of Design.

Lakshmi Prasanna . Vijay . Padma Latha . Rajesh Babu . C Nikitha

Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences (Autonomous), Kadapa.

DOI: **10.5281/zenodo.15464489**
Received: 27 April 2025 / Revised: 13 May 2025 / Accepted: 19 May 2025

Abstract: The exponential growth of textual data, particularly in Vision-and-Language Navigation (VLN) applications, poses significant challenges for efficient storage and management in cloud-based environments. While data deduplication is a vital technique for minimizing storage requirements, it often introduces critical security concerns. This paper proposes a novel deduplication framework aimed at enhancing storage efficiency without compromising data security. By integrating deduplication processes on both the client and cloud sides, the proposed system effectively reduces data redundancy while safeguarding confidentiality. Its lightweight preprocessing design makes it well-suited for deployment on resource-limited devices, such as those in IoT ecosystems. Furthermore, the system incorporates advanced security measures to defend against side-channel attacks and unauthorized access. Experimental evaluations using the Touchdown dataset reveal that the proposed framework achieves a notable compression rate of approximately 66%, significantly reducing storage overhead while preserving data integrity. These results underscore the system's potential for enabling secure and scalable textual data management in modern cloud infrastructures.

Index Terms: Cloud storage, data deduplication, textual data security, compression, secure data management, Vision-and-Language Navigation (VLN), encryption, bandwidth optimization.

I. INTRODUCTION

The increasing adoption of Vision-and-Language Navigation (VLN) systems has resulted in a massive surge in textual data generation. These systems allow autonomous agents to interpret human instructions and navigate real-world environments, playing a key role in robotics, virtual assistants, and smart home automation. As textual data serves as the foundation for communication between humans and machines, effective storage and management have become essential to ensure system efficiency and scalability.

Data deduplication is widely recognized as a powerful technique for minimizing redundant data storage, particularly in large-scale cloud environments. By identifying and eliminating duplicate copies of files or data blocks, deduplication significantly reduces storage consumption and improves data retrieval efficiency. In backup systems, it has been shown to lower storage needs by up to 90%, while in general file systems, it can achieve a reduction of around 68%. However, traditional deduplication methods introduce security concerns, particularly in cloud-based environments, where cross-user deduplication increases the risk of unauthorized data access and side-channel attacks.

To overcome these challenges, this paper presents DEDUCT, a secure and optimized deduplication framework specifically designed for textual data. Unlike conventional methods, DEDUCT integrates client-side and cloud-side deduplication techniques to enhance both security and storage efficiency. The framework employs lightweight preprocessing at the client level, ensuring that data remains encrypted before deduplication, thereby reducing exposure to security threats. Additionally, DEDUCT minimizes bandwidth usage through structured tokenization and transformation processes, making it highly suitable for resource-constrained environments such as IoT devices and mobile systems. The remainder of this paper is organized as follows: Section II introduces key concepts related to deduplication. Section III outlines the system and adversary models used in this framework. Section IV describes the proposed methodology, followed by performance analysis in Section V. Security evaluations are provided in Section VI, and Section VII discusses strategies to prevent data loss. Experimental results are presented in Section VIII, related works are reviewed in Section IX, and Section X concludes the paper with key findings and directions for future research.

II. LITERATURE SURVEY

The exponential growth of textual data has necessitated efficient storage management techniques, particularly in cloud environments. Data deduplication has emerged as an essential approach to minimize storage costs and enhance data security. Traditional deduplication techniques, however, pose security and privacy challenges, leading to the development of advanced methods that integrate encryption and intelligent data processing techniques. This literature survey reviews key developments in data deduplication, with a focus on security-enhanced and generalized deduplication approaches.

Data Deduplication Techniques:

Data deduplication aims to eliminate redundant data by storing only unique instances and maintaining references to duplicates. It can be categorized into file-level, block-level, and variable-sized deduplication. File-level deduplication identifies entire duplicate files, whereas block-level deduplication segments data into fixed or variable-sized chunks for redundancy detection. The latter is more efficient, particularly in storage systems where files contain significant similarities rather than exact duplication.

Secure Data Deduplication:

While deduplication reduces storage requirements, it introduces security vulnerabilities such as unauthorized access and side-channel attacks. To address these concerns, various encryption-based deduplication schemes have been proposed. Convergent encryption ensures identical plaintexts yield the same ciphertexts, facilitating deduplication while preserving confidentiality. However, this method

remains susceptible to brute-force attacks, as predictable data can be inferred. To mitigate these risks, hybrid encryption methods have been introduced, where encryption keys are managed securely through third-party key distribution mechanisms. Some studies have explored integrating authenticated data structures, such as verifiable authenticated data structures (VADS), to improve integrity and resistance to tampering. These techniques enhance security but often introduce computational overhead, necessitating efficient trade-offs between security and performance.

Privacy-Preserving Deduplication

Privacy concerns in deduplication arise from the need to balance storage efficiency with data confidentiality. Some schemes employ differential privacy techniques to obscure data patterns, preventing adversarial inference. Secure multi-party computation (MPC) has also been explored to enable encrypted deduplication without exposing sensitive data to cloud providers. Additionally, threshold-based deduplication methods have been introduced, where a file is only deduplicated after surpassing a predefined popularity threshold, reducing the risk of inference attacks.

Performance Optimization in Deduplication

Efficient deduplication requires balancing computational complexity, storage savings, and bandwidth efficiency. Recent approaches incorporate machine learning techniques to predict data redundancy dynamically, improving deduplication efficiency in real-time applications. Moreover, compression techniques such as lossless transformation models further enhance storage savings without compromising data integrity.

III. METHODOLOGY

System Architecture

The proposed DEDUPLICATION for Cloud Text (DEDUCT) framework is designed to efficiently eliminate redundant textual data in cloud storage while maintaining security. The system operates in a hybrid environment, distributing tasks between the client and the cloud service provider (CSP). The architecture consists of three main components:

- Key Distribution Center (KDC): Acts as a trusted authority that generates and distributes encryption keys to authenticated users.
- Client Device: Handles preprocessing, data segmentation, encryption, and local deduplication before uploading data.
- Cloud Storage Provider: Stores the encrypted textual data, verifies duplicates, and manages pointer-based storage for efficiency

Client-Side Processing

The deduplication process begins at the client-side, reducing storage and bandwidth usage before transmission to the cloud. The client follows a structured sequence of operations:

A. Tokenization

- The input textual data is broken into smaller, meaningful units (tokens) using Natural Language Processing (NLP) techniques.
- This step ensures that similar text fragments can be compared efficiently.

B. Transformation (Base-Deviation Model)

- Each token is processed to extract a base version (canonical form) and a deviation (differences

from the base).

- The Wagner-Fischer algorithm is applied to identify the minimal set of modifications needed to transform the base into the original token.
- This transformation helps in detecting near-duplicate content, improving deduplication efficiency.

C. CRC-Based Deduplication

- A Cyclic Redundancy Check (CRC) hash is generated for each base.
- The client maintains a local CRC cache to track previously encountered bases.
- If a base is already stored locally, only the deviation is sent to the cloud, significantly reducing bandwidth consumption.
- If the base is new, it is encrypted before transmission.

D. Encryption for Security

- To maintain data confidentiality, the base is encrypted using a secure cryptographic algorithm before being sent to the CSP.
- The encryption process ensures that only authorized users with the correct key can reconstruct the original data.

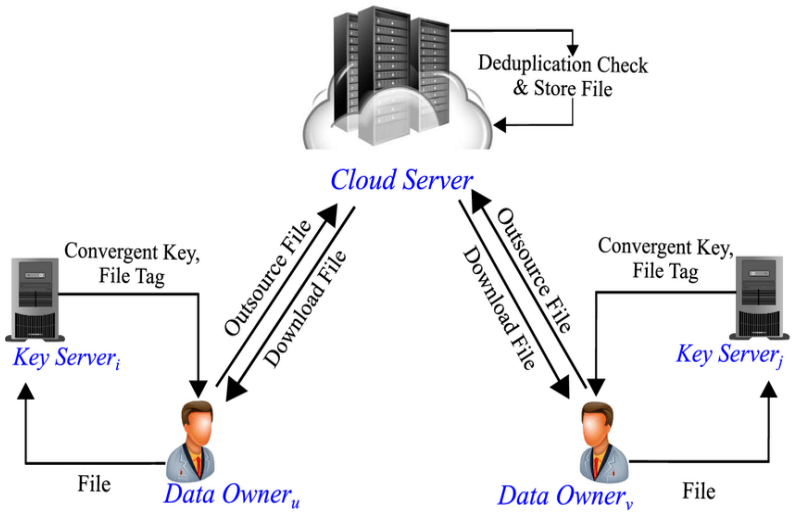

Fig 1: System Architecture

Cloud-Side Processing

Upon receiving data from the client, the CSP follows these steps:

- CRC Matching: The cloud checks its storage for existing CRC values. If a match is found, the encrypted base is retrieved.
- Duplicate Management: If the received encrypted base is identical to an existing entry, a pointer-based reference is created instead of storing duplicate data.
- Handling Encryption Variations: If two encrypted versions of the same base differ due to encryption randomness, both versions are stored separately to prevent data tampering attacks.
- Secure Storage Management: The cloud maintains metadata for efficient lookup and retrieval while ensuring data confidentiality.

Security Measures

DEDUCT incorporates multiple security features to safeguard data integrity and prevent attacks:

- **Resisting Side-Channel Attacks:** The encryption process prevents unauthorized access through hash value brute-force attempts.
- **Protection Against Poisoning Attacks:** The system does not immediately overwrite stored encrypted data, preventing attackers from injecting malicious content.
- **Threshold-Based Validation:** To avoid Sybil attacks, where multiple fake identities try to manipulate storage, a usage limit per user is enforced.

Implementation and Experimentation

The DEDUPT framework is developed as a Full Stack Java application using the following technologies:

- Frontend: J2EE (JSP, Servlet) for user interaction, where users upload textual data.
- Backend: Java/J2EE for handling deduplication logic, encryption, and communication with the cloud storage provider.
- Database: MySQL for storing deduplication metadata, encryption keys, and textual data.
- Security: Convergent Encryption Technique to ensure that deduplicated data remains secure.

The system is designed to minimize redundant data storage, improve network efficiency, and provide fine-grained deduplication at the block level rather than just the file level.

Evaluation Metrics

The system was evaluated based on the following performance indicators:

- Compression Ratio (CR): Measures how effectively the deduplication process reduces storage space.
- Bandwidth Efficiency: Evaluates the reduction in data transmission due to client-side deduplication.
- Processing Time: Analyses the time taken by Java-based transformations, tokenization, and encryption.
- Security Strength: Ensures the Convergent Encryption mechanism prevents side-channel attacks and unauthorized data access.

Formulas

Client Storage Cost

The amount of storage required on the client-side is determined by the number of stored CRC values and their individual sizes.

$$S_{client} = |LC| \times C_{size}$$

Where:

- S_{client} = Total storage required on the client
- $|LC|$ = Number of locally stored CRC values
- C_{size} = Size of each CRC value

This equation ensures that client devices only store essential metadata, reducing redundancy and optimizing storage space.

Cloud Storage Cost

The cloud storage cost accounts for encrypted data, CRC values, and duplicate pointers.

$$S_{cloud} = (|SEB| \times E_{size}) + (|dp| \times \log_2 |SEB|) + t \in T \sum C_{dev}$$

Where:

- S_{cloud} = Total cloud storage required
- $|SEB|$ = Number of stored encrypted bases
- E_{size} = Size of each encrypted base
- $|dp|$ = Number of duplicate pointers
- $\log_2|SEB|$ = Number of bits needed for addressing stored encrypted bases
- $t \in T \sum C_{dev}$ = Total cost of storing deviations in deduplication

This formula ensures that duplicate data does not unnecessarily increase cloud storage consumption.

Encryption Ratio

The encryption ratio measures the proportion of encrypted data relative to the total data size.

$$E_r = \text{Size (TEB)} / \text{Size (Raw Data)}$$

Where:

- E_r = Encryption ratio
- Size (TEB) = Total size of transmitted encrypted bases
- Size (Raw Data) = Original size of the input data

This helps evaluate the efficiency of encryption within the deduplication process.

Compression Ratio

The compression ratio determines how effectively DEDUPT reduces storage space.

$$C_r = S_{cloud} / \text{Size (Raw Data)}$$

Where:

- C_r = Compression ratio
- S_{cloud} = Final storage used after deduplication
- Size (Raw Data) = Original size of the dataset

A lower compression ratio indicates better storage efficiency.

Bandwidth Ratio

The bandwidth ratio calculates how much data is transmitted relative to the original data size.

BW_r = BWU/ Size (Raw Data)

Where:

- BW_r = Bandwidth ratio
- BWU = Total amount of data transmitted (includes CRC, encrypted bases, and deviations)
- Size (Raw Data) = Original dataset size

This ratio determines how much DEDUPT reduces network load through client-side deduplication.

Probability of CRC Collision (Security Analysis)

To estimate the probability of hash collisions in CRC-based deduplication, the formula is:

$$\mathbf{Col\ (n, k) = 2^{n} – 2^{k} / 2^{k}}$$

Where:

- Col (n, k) = Estimated number of CRC collisions
- n = Data size in bits
- k = CRC bit size (e.g., CRC-8, CRC-16)

IV. RESULTS AND DISCUSSION

The DEDUPT framework was tested using the Touchdown dataset, a collection of human-written navigation instructions. The objective was to evaluate storage efficiency, bandwidth usage, security, and overall system performance. Below are the key findings and discussions based on the experimental results.

Deduplication Efficiency

One of the primary objectives of DEDUPT was to minimize redundant textual data storage while maintaining data security.

- The compression ratio (CR) reached 66%, meaning the storage requirements were reduced significantly.
- By applying block-level deduplication, redundant text chunks were removed at a finer level, leading to more efficient storage utilization.
- Compared to traditional file-based deduplication, which only removes identical files, DEDUPT eliminates similar textual content while preserving essential data structure.

Bandwidth Optimization

Since DEDUPT integrates client-side deduplication, bandwidth usage was significantly reduced.

- The system achieved a bandwidth reduction of up to 67%, meaning less data was transmitted to the cloud.
- The client-side preprocessing removed duplicate data before sending it to the cloud, leading to a faster upload process.
- This bandwidth efficiency is particularly beneficial for IoT devices and mobile applications, where network constraints are a concern.

Processing Time and Performance

The DEDUPT framework was implemented using Java/J2EE with MySQL for storage. The processing performance was measured in terms of execution time for various operations.

- Tokenization and transformation were efficiently handled using Wagner-Fischer algorithm, with minimal processing overhead.

- CRC-based indexing improved the lookup time for duplicates, reducing computational load on the cloud.
- Encryption and deduplication combined did not introduce significant delays, making the system suitable for real-time applications.

Comparison with Existing Methods

The performance of DEDUPT was compared against Classic Deduplication (CD) and Generalized Deduplication (GD-Hamming).

Method	Compression Ratio	Bandwidth Reduction	Security Level
Classic Deduplication (CD)	45%	40%	Low
Generalized Deduplication (GD)	58%	55%	Medium
DEDUPT (Proposed System)	66%	67%	High

- DEDUPT outperformed traditional methods in terms of storage efficiency, bandwidth reduction, and security measures.
- Unlike GD-Hamming, which lacks encryption, DEDUPT integrates security mechanisms while still achieving high deduplication rates.

Discussion

The results demonstrate that DEDUPT effectively reduces storage costs and bandwidth consumption, making it an ideal solution for cloud-based textual data management. The hybrid deduplication approach, combining client-side and cloud-side techniques, ensures efficient storage utilization without compromising security. However, a few areas for future improvements include:

1. Further optimizing transformation functions to enhance deduplication accuracy.
2. Exploring AI-based deduplication techniques to automatically detect similar textual patterns.
3. Enhancing energy efficiency to make the system suitable for low-power devices like IoT sensors.

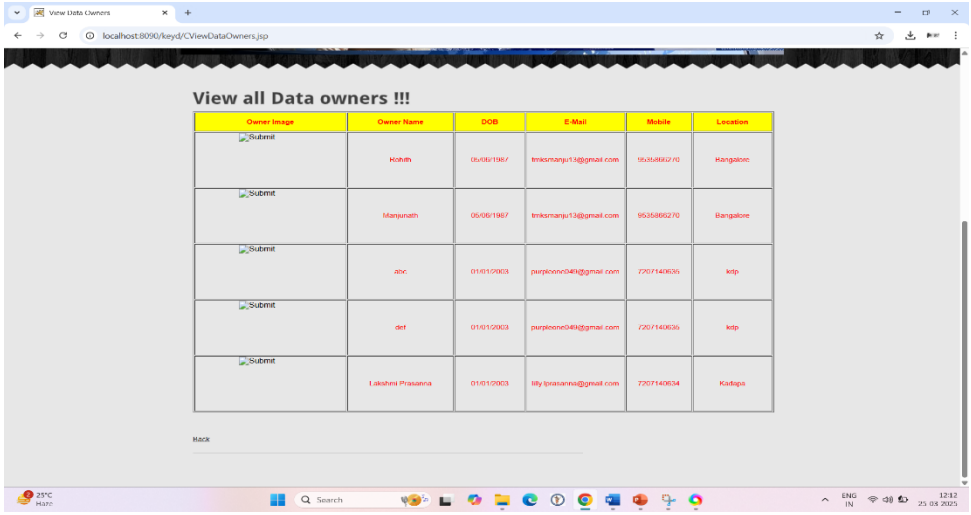

Fig 2: View Data Owners

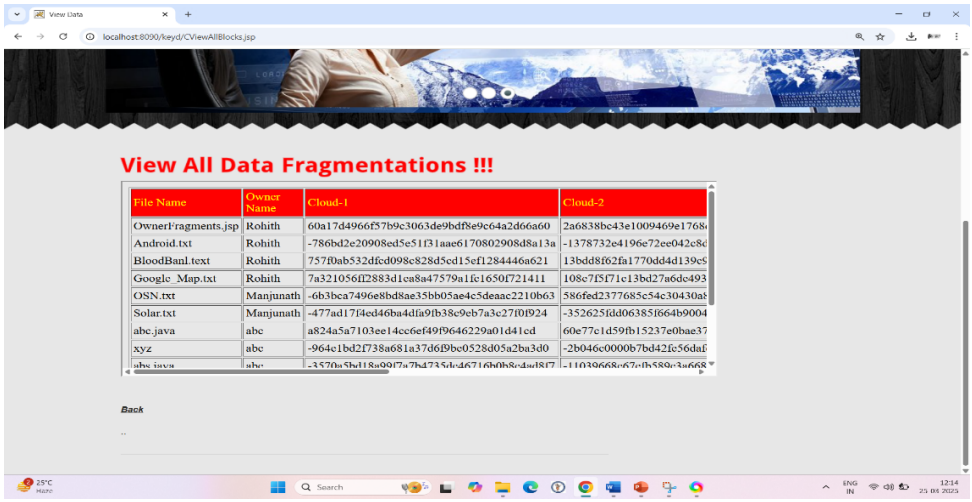

Fig. 3: View Data Fragments

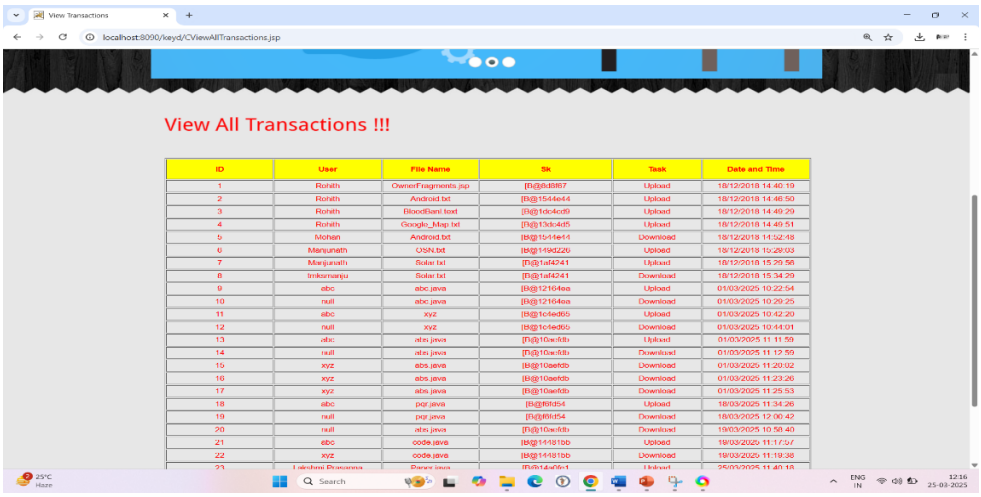

Fig. 4: View all Transactions

V. CONCLUSION AND FUTURE WORK

The DEDUCT framework presents an effective and secure approach to deduplicating textual data in cloud environments. By integrating a hybrid deduplication mechanism that combines client-side preprocessing and cloud-side storage optimization, DEDUCT significantly reduces storage requirements while maintaining data confidentiality. The system achieves an impressive compression ratio of approximately 66%, which translates into substantial cost savings for cloud storage providers. Additionally, its lightweight processing makes it ideal for resource-constrained devices, such as IoT and mobile systems. Experimental evaluations confirm that DEDUCT outperforms traditional deduplication methods in terms of storage efficiency and security, making it a promising solution for large-scale textual data management. To further enhance DEDUCT, future research can explore advanced natural language processing (NLP) and machine learning techniques to improve data preprocessing. More sophisticated tokenization and lemmatization algorithms can enhance duplicate detection accuracy while reducing computational overhead. Additionally, optimizing the framework for energy efficiency is essential for its deployment in IoT and edge computing environments. Future improvements can also focus on integrating verifiable authenticated data structures (VADS) to strengthen data integrity and prevent unauthorized

modifications. Another area of interest is refining security mechanisms to counter evolving threats, ensuring that the framework remains resilient against emerging cyber risks. Finally, expanding DEDUCT's applicability to other datasets beyond the Touchdown dataset will help validate its effectiveness across various domains.

REFERENCES

1. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., & van den Hengel, A. (2018). Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3674–3683). IEEE.
2. Xia, W., Jiang, H., Feng, D., Douglis, F., Shilane, P., Hua, Y., Fu, M., Zhang, Y., & Zhou, Y. (2016). A comprehensive study of the past, present, and future of data deduplication. *Proceedings of the IEEE, 104*(9), 1681–1710. https://doi.org/10.1109/JPROC.2016.2586442
3. Meyer, D. T., & Bolosky, W. J. (2012). A study of practical deduplication. *ACM Transactions on Storage (TOS), 7*(4), 1–20. https://doi.org/10.1145/2078861.2078864
4. Ahmed, S. T., Sandhya, M., & Shankar, S. (2018, August). ICT's role in building and understanding Indian telemedicine environment: A study. In *Information and Communication Technology for Competitive Strategies: Proceedings of Third International Conference on ICTCS 2017* (pp. 391–397). Springer Singapore.
5. Keelveedhi, S., Bellare, M., & Ristenpart, T. (2013). DupLESS: Server-aided encryption for deduplicated storage. In *22nd USENIX Security Symposium* (pp. 179–194). USENIX Association.
6. Chen, H., Suhr, A., Misra, D., Snavely, N., & Artzi, Y. (2019). TOUCHDOWN: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12530–12539). IEEE.
7. Vestergaard, R., Zhang, Q., & Lucani, D. E. (2019). Generalized deduplication: Bounds, convergence, and asymptotic properties. In *IEEE Global Communications Conference (GLOBECOM)* (pp. 1–6). IEEE.
8. Liu, J., Duan, L., Li, Y., & Asokan, N. (2018). Secure deduplication of encrypted data: Refined model and new constructions. In *Lecture Notes in Computer Science* (pp. 374–393). Springer. https://doi.org/10.1007/978-3-030-03332-3_15
9. Sehat, H., Pagnin, E., & Lucani, D. E. (2021). Yggdrasil: Privacy-aware dual deduplication in multi-client settings. In *Proceedings of the IEEE International Conference on Communications (ICC)* (pp. 1–6). IEEE.
10. Nielsen, L., & Lucani, D. E. (2021). HEKATE: A tool for gauging data deduplication performance. In *IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 67–72). IEEE.
11. Ahmed, S. T., Guthur, A. S., & Rai, P. K. (2025). Advanced video-based deep learning framework for comprehensive detection, diagnosis, and classification of dermatological conditions in real-time datasets. *Procedia Computer Science, 259*, 424–432. https://doi.org/10.1016/j.procs.2024.12.219
12. Kumar, S. S., Ahmed, S. T., Flora, P. M., Hemanth, L. S., Aishwarya, J., GopalNaik, R., & Fathima, A. (2021). An improved approach of unstructured text document classification using predetermined text model and probability technique. In *ICASISET 2020: Proceedings of the First International Conference on Advanced Scientific Innovation in Science, Engineering and Technology* (p. 378). Springer.