Data Compression with Backbone Network

MD FAISAL
Department of Computer Science & Engineering,
Raj Kumar Goel Institute of Technology,
Ghaziabad, UP, India
mdfaisalmallick0786@gmail.com

SAKIL AHAMD
Department of Computer Science & Engineering,
Raj Kumar Goel Institute of Technology,
Ghaziabad, UP, India
itm.shakil25@gmail.com

MOHD SHARJIL *Department of Computer Science & Engineering,*
Raj Kumar Goel Institute of Technology,
Ghaziabad, UP, India
Sharjilansari234@gmail.com

REYAJUL ALAM
Department of Computer Science & Engineering, Raj Kumar Goel Institute of Technology,
Ghaziabad, UP, India
reyajulalam21@gmail.com

MOHD.ARSH JUNAID
Department of Computer Science & Engineering, Raj Kumar Goel Institute of Technology,
Ghaziabad, UP, India
mohdarshjunaid@gmail.com

Abstract— **Blockchain technology is increasingly recognized as a transformative innovation that is reshaping how society and individuals interact. Following the invention of the Internet, which broke down communication barriers between countries, blockchain has shifted the concept of trust from centralized third parties to decentralized protocols.**

Blockchain technology enables the creation of decentralized ecosystems where transactions can be exchanged securely. Characteristics like persistence, anonymity, and auditability are embedded in the protocol's design.

However, despite the enthusiasm surrounding blockchain, major platforms such as Bitcoin and Ethereum have encountered significant challenges. Bitcoin's blockchain size has reached approximately 250 GB, while Ethereum's is nearing 1 TB. This massive data volume must be replicated across all users who wish to participate in the network.

Currently, no technology exists to effectively reduce the disk space required by blockchain applications. Moreover, there is no lightweight protocol that allows users to securely and privately participate in these networks.

As the world moves toward a future dominated by resource-constrained devices such as smartphones and IoT devices, reducing the disk space required for blockchain is crucial. This advancement is a key factor in enabling the widespread deployment of this groundbreaking technology.

Keywords— Compression Algorithms, Blockchain Technology, Decentralized Ecosystem, Distributed Storage, Lossless Compression, Lossy Compression, Smart Contracts, Cryptographic Hashing, Consensus, Protocols, Data Integrity, Scalability

I. INTRODUCTION

Data compression and blockchain technology have emerged as transformative forces in the digital landscape. Data compression aims to reduce the size of digital information, optimizing storage and transmission, while blockchain ensures a decentralized, secure, and tamper-proof ledger system. Combining these technologies can enhance the scalability, security, and efficiency of data management, addressing critical challenges faced by industries reliant on large-scale data handling. This paper explores the integration of blockchain as a backbone network for data compression, highlighting its potential to revolutionize data storage and transfer mechanisms.

The exponential growth of data in the information age presents significant challenges, including limited storage capacity and constrained network resources. Blockchain technology, recognized for its decentralized and secure nature, is gaining prominence across industries. However, blockchain networks like Bitcoin and Ethereum require vast storage capacities, with data sizes reaching hundreds of gigabytes. Simultaneously, compression techniques have been developed to optimize storage and data transmission by reducing file sizes. Combining blockchain with advanced compression techniques presents an opportunity to create an efficient and scalable ecosystem, addressing the storage concerns associated with decentralized networks.

.

II. Literature Survey

Various studies have emphasized the need for efficient data management strategies in an era of exponential data growth. Compression techniques, ranging from lossless methods like Huffman encoding and LZ77 to lossy methods for multimedia data, have been extensively developed. Concurrently, blockchain technology, initially popularized by cryptocurrencies like Bitcoin, has gained traction in sectors such as healthcare, supply chain, and finance. Recent works have explored blockchain's ability to handle large datasets through innovations like pruning modes, light clients, and segmentation.

Efforts to compress blockchain data have also been documented. Techniques like streaming compression and the mini-blockchain scheme aim to reduce the size of blockchain databases, enabling their adoption in resource-constrained environments such as IoT. However, these approaches often compromise certain features, such as real-time data access or verification capabilities. Integrating robust compression techniques with blockchain can offer a holistic solution, enhancing both storage efficiency and system integrity.

Existing research highlights the potential of blockchain to ensure data security, immutability, and transparency. Studies have shown that blockchain applications face scalability challenges due to their extensive storage requirements. Concurrently, compression algorithms have been extensively researched, categorized into lossy and lossless techniques. Lossless compression ensures the complete reconstruction of original data, making it suitable for documents and code, while lossy compression is effective for multimedia applications. However, integrating compression into blockchain protocols remains largely unexplored, with limited frameworks addressing both storage optimization and the decentralized nature of blockchain systems.

Table I. COMPARATIVE ANALYSIS OF EXISTING RESEARCH ON RECOMMENDATION MODELS

Author	**Research Objective**	**Methodology**	**Key Findings**
Ayub et al. [7], (2019)	Performance analysis of blockchain-based data compression models for network optimization	Blockchain-based compression techniques, such as IPWR method	Blockchain integration significantly improves data compression and network performance, reducing latency and enhancing security.
Adomavicius et al. [10], (2005)	Deliver a comprehensive summary of cutting-edge blockchain-based data compression methods and explore future extensions	Blockchain, cryptographic techniques, and statistical methods	Blockchain enhances data compression by improving forecast accuracy and ensuring secure data transmission over networks.
Ahuja R et al. [12], (2019)	Propose a model using blockchain for secure and efficient data compression in network environments	Blockchain-based compression model, cryptographic techniques	Blockchain-based compression reduces data transfer times and lowers RMSE, improving network performance.
Koren Y et al. [15], (2010)	Analysis of the role of trust in blockchain-enabled data compression systems	Trust metrics in blockchain-based networks	Trust-based systems significantly improve data compression accuracy and efficiency in blockchain-enabled networks, especially with sparse data interactions.
Isinkaye et al. [1], (2015)	A comparison of blockchain-based data compression models and traditional methods	Blockchain-Data Integration, User-Item Association	Blockchain integration provides enhanced data security, reducing redundancy and ensuring efficient compression even with large data volumes.
Kassak et al. [4], (2016)	Hybrid blockchain and data compression strategies for optimized network throughput	Hybrid blockchain model for data compression	Hybrid blockchain-based compression outperforms traditional compression techniques by reducing data overhead and increasing throughput.

III. Technology Used

A data compression method using blockchain [18] integrates decentralized validation with advanced compression algorithms. The basic principle of blockchain-based compression is that a data packet is compressed using an algorithm that ensures data integrity and security by validating the compressed data across multiple nodes in the blockchain network.According to Figure 1, the data packet is compressed and verified across the network, where each node validates the compressed data, ensuring its consistency, while the original data is stored on the blockchain for transparency and tamper-resistance.**B. Blockchain Compression With Cryptographic Means**In the Blockchain Compression With Cryptographic Means method [19], cryptographic techniques are applied to the data before compression to ensure additional security. This method uses digital signatures to validate the integrity of the compressed data, ensuring that any modifications to the compressed data are traceable.

On the other hand, **Lossy Compression** algorithms, like **JPEG** for images and **MP3** for audio, achieve higher compression ratios by eliminating some of the data that may not be perceptible to human senses. These techniques work by removing less significant information, ensuring that the data still appears nearly identical to the original but with a reduced size.**Dictionary-based Compression** methods, such as **LZ77** and **LZ78**, use a dictionary to replace repeated sequences of data with shorter references. These methods are particularly effective for compressing text and other forms of structured data.In the realm of multimedia, **Wavelet Transform Compression** is frequently used in images and video compression. It operates by transforming the image data into frequency components, selectively discarding high-frequency details that are less noticeable to the human eye, similar to how JPEG operates but with improved quality and efficiency in certain cases. Moreover, **Machine Learning** is being integrated into modern data compression techniques. Through pattern recognition and optimization algorithms, machine learning models can predict which parts of the data can be compressed the most, dynamically adapting to different data types for optimal compression performance.**Parallel and Distributed Compression** techniques are also evolving, where the compression process is performed in parallel across multiple machines or cores. This is particularly useful when dealing with large-scale datasets or real-time data compression, such as in cloud-based storage systems, network data transmission, and video streaming

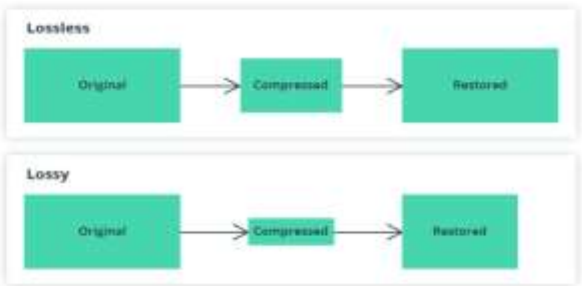


Fig. 1 example of compression algorithm

A. *Lossless Data Compression Techniques*

In the Lossless Data Compression technique [20], the original data is encoded in such a way that it can be fully reconstructed without any loss. One common method used is Huffman coding, where frequent elements are assigned shorter codes and less frequent elements are assigned longer codes. The encoded data can be decompressed back to its original form with no data loss.The basic idea behind Huffman coding is to assign variable-length codes to input characters, with shorter codes assigned to more frequent characters. The formula to compute the optimal code length for each symbol is:

$$L(s) = -\log 2(p(s))$$

Where L(s) represents the length of the code for symbol s and p(s) represents the probability of symbol s occurring in the dataset.

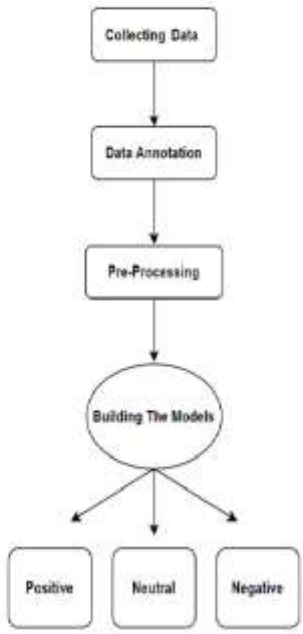
B. *Similarity Measures*

To measure the similarity between datasets or compressibility, we often use techniques like **Euclidean distance** or **Cosine Similarity**. For data compression, Cosine Similarity can be applied to measure how similar two datasets are based on their vector representations. This helps in determining which portions of the data can be compressed more effectively. The formula for Cosine Similarity is:

$$\text{Cosine Similarity} = \frac{\sum a_i \cdot b_i}{\sqrt{\sum a_i^2} \cdot \sqrt{\sum b_i^2}}$$

Where ai and bi are the components of two vectors, and the result ranges between 0 (no similarity) and 1 (identical)..

IV. PROPOSED WORK



The proposed framework introduces a comprehensive integration of blockchain technology with a layered data compression model designed to optimize data storage and retrieval processes. In the **Data Ingestion Layer**, raw data sourced from a wide range of origins, such as IoT devices, enterprise systems, and cloud platforms, is collected and processed. This data is then segmented into manageable chunks, which ensures more efficient handling and better scalability for compression

Fig. 2. Workflow of the Movie recommendation system.

A. *Pseudocode for Data Compression for Backbone Network Using Blockchain:Store the instructions and examine the data*

I. Store the instructions and examine the data

II. Select an appropriate compression algorithm based on data type (lossy or lossless)

III. For each data chunk:

- Compress the data using the selected compression algorithm
- Generate a cryptographic hash of the compressed data
- Store the compressed data and hash on the blockchain
- Ensure data integrity by linking the compressed data using Merkle trees

IV. When retrieving data:

- Retrieve the compressed data chunk and its hash from the blockchain

- Verify the integrity of the data using the cryptographic hash

- Decompress the data using the appropriate decompression algorithm

V. Finish

B. Implementation

The data compression structure for backbone networks is being developed using blockchain technology to ensure data integrity, security, and efficient storage. The implementation uses **lossless or lossy compression algorithms** depending on the data type. The compressed data is securely stored on the blockchain, and each data chunk is hashed using a cryptographic hash function. This ensures that data integrity is maintained, and no tampering occurs during storage and retrieval. The blockchain also utilizes **Merkle trees** for efficient linking of data chunks.

The process of data compression for a backbone network using blockchain involves several key steps. First, essential libraries such as **Pandas**, **NumPy**, **Scipy**, and **PyCrypto** are imported to handle data manipulation, compression, cryptographic hashing, and blockchain integration. The raw data, which could include IoT sensor data, multimedia files, or transactional records, is loaded into a Jupyter environment using the **Pandas** library for easy processing and analysis. A feature matrix is then created, where rows represent data chunks and columns represent the data's features. This matrix is converted into a sparse matrix using **Scipy**, optimizing storage by only retaining non-zero values, thus reducing space requirements. Next, an appropriate **compression algorithm**—either lossy or lossless—is applied depending on the data type. The compressed data chunks are stored on the blockchain, where each chunk is hashed using a cryptographic hash function, ensuring data integrity and security. The blockchain's **Merkle tree** structure helps link the data chunks, making it easier to manage and retrieve them while maintaining consistency. When the data is needed, it is retrieved from the blockchain, and its integrity is validated using the cryptographic hash. The data is then decompressed using the correct decompression algorithm, ensuring that only authorized users can access the data. This entire process ensures efficient data compression, secure storage, and quick access to compressed data, all while maintaining high levels of transparency and data integrity..

V. Conclusion and Future Scope

Integrating blockchain with data compression presents a promising approach to efficient and secure data management. The proposed framework leverages advanced compression algorithms, cryptographic techniques, and blockchain's decentralized nature to address scalability and data integrity challenges. By enabling resource-constrained devices, such as IoT systems, to participate in blockchain networks, this approach expands blockchain's applicability to diverse industries.

Future research could focus on optimizing compression algorithms for specific data types and exploring hybrid blockchain architectures to balance decentralization and efficiency. Additionally, integrating AI-driven predictive models for dynamic compression and storage optimization could further enhance the system's performance. As data generation continues to surge, this innovative approach could serve as a cornerstone for sustainable and secure data ecosystems.

References

[1]. Isinkaye F, Folajimi Y, Ojokoh B (2015) “Recommendation systems: principles, methods and evaluation”, Egypt Inform J 16(3):261–273. https://doi.org/10.1016/j.eij.2015.06.005

[2]. Pazzani MJ (1999) A framework for collaborative, content-based and demographic filtering. Artif Intell Rev 13(5–6):393–408. https://doi.org/10.1023/A:1006544522159

[3]. Garg H, Singh M, Sharma V, Agarwal M (2022) "Decentralized Application (DAPP) to enable E-voting system using Blockchain Technology," 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, pp. 1-6, doi: 10.1109/ICCSEA54677.2022.9936413.

[4]. Kassak O, Kompan M, Bieliková M (2016) Personalized hybrid recommendation for group of users: Top-N multimedia recommender. Information Processing and Management, 52(3), 459–477. https://doi.org/10.1016/j.ipm.2015.10.001

[5]. Sarwar BM, Karypis G, Konstan J, Riedl J (2002) Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In: Proceedings of the fifth international conference on computer and information technology, vol 1, pp 291–324.

[6]. Shardanand U, Maes P (1995) Social information filtering: algorithms for automating word of mouth. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp 210–217. https://doi.org/10.1145/223904.223931

[7]. Ayub M, Ghazanfar MA, Mehmood Z, Saba T, Alharbey R, Munshi AM, Alrige MA (2019) Modeling user rating preference behavior to improve the performance of the collaborative filtering-based recommender systems. PLoS ONE 14(8):e0220129. https://doi.org/10.1371/journal.pone.0220129

[8]. Singh, R., Sharma, R., Kumar, K., Singh, M., Vajpayee, P. (2024). Securing lives and assets: IoT-Based earthquake and fire detection for Real-Time monitoring and safety. In Communications in computer and information science (pp. 15–25). https://doi.org/10.1007/978-3-031-56703-2_2

[9]. Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. In: The adaptive web, Springer, pp 291–324.

[10]. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749. https://doi.org/10.1109/TKDE.2005.99

[11]. Gupta, Megha; Singh, Mandeep; Sharma, Anupam; Sukhija, Namrata; Aggarwal, Puneet Kumar; Jain, Parita: 'Unification of machine learning and blockchain technology in healthcare industry' (Healthcare Technologies, 2023), 'Innovations in Healthcare Informatics: From interoperability to data analysis', Chap. 6, pp. 185-206, DOI: 10.1049/PBHE041E_ch6, IET Digital Library, https://digital-library.theiet.org/content/books/10.1049/pbhe041e_ch6

[12]. Ahuja R, Solanki A, Nayyar A (2019) Movie recommender system using k-means clustering and k-nearest neighbor. In: 2019 9th international conference on cloud computing, data science & engineering (confluence), IEEE, pp 263–268. https://doi.org/10.1109/CONFLUENCE.2019.8776969

[13]. Makar K, Goel S, Kaur P, Singh M, Jain P, Aggarwal PK (2021) Reliability of Mobile Applications: A Review and Some Perspectives, 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-4, doi: 10.1109/ICRITO51393.2021.9596350.

[14]. Halder S, Sarkar AMJ, Lee YK (2012) Movie recommendation system based on movie swarm. In: 2012 Second international conference on cloud and green computing, pp 804–809. https://doi.org/10.1109/CGC.2012.121

[15]. Koren Y (2010) Factor in the neighbors: scalable and accurate collaborative filtering. ACM Trans Knowl Discov Data (TKDD) 4(1):1–24. https://doi.org/10.1145/1644873.1644874

[16]. Goel A, Singh M, Gupta J, Mangla N (2023) Med Card: an innovative way to keep your medical records handy and safe. In Advances in intelligent systems and computing (pp. 51–60). https://doi.org/10.1007/978-981-99-0550-8_4

[17]. Bahadorpour M, Neysiani BS, Shahraki MN (2017) Determining optimal number of neighbors in item-based knn collaborative filtering algorithm for learning preferences of new users. J Telecommun Electron Comput Eng (JTEC) 9(3):163–167.

[18]. Singh, M., & Malik, A. (2024). Multi-hop routing protocol in SDN-based wireless sensor network: a comprehensive survey. In Software-Defined Network Frameworks, (1st ed., pp. 121-141.). CRC Press.

[19]. Chaudhary P, Goel S, Jain P, Singh M, Aggarwal PK, Anupam (2021) "The Astounding Relationship: Middleware, Frameworks, and API," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021, pp. 1-4, doi: 10.1109/ICRITO51393.2021.9596088.

[20]. Singh, M., Chauhan, N., Verma, A., Parasher, A., Budhiraja, G. (2019). Implementation of database using python flask framework. International Journal of Engineering and Computer Science, 8(12), 24894–24899. https://doi.org/10.18535/ijecs/v8i12.4390

[21]. Rajput R, Singh M, Srivastava Y, Srivastava P, Parihar N (2023) "Decentralized Finance App – Tip Wallet", TIJER - International Research Journal, Vol.10, Issue 5, page 58-62. Available: http://www.tijer.org/papers/TIJER2305128.pdf

[22]. Airen S, Agrawal J (2022) Movie Recommender System Using K-Nearest Neighbors Variants. Natl. Acad. Sci. Lett. 45, 75–82. https://doi.org/10.1007/s40009-021-01051-0