International Journal of Advanced Technology and Engineering Exploration, Vol 11(115)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2023.10102422

Review Article

Smart storage strategies for blockchain: a review of approaches

Shelke R. Kavita[1*] and Subhash K. Shinde[2]
Assistant Professor, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Navi Mumbai, Maharashtra, India[1]
Professor, Department of Computer Engineering, Lokmanya Tilak College of Engineering, Koparkhairane, Navi Mumbai, Maharashtra, India[2]

Received: 15-October-2023; Revised: 13-June-2024; Accepted: 14-June-2024

Abstract
Blockchain technology has drawn the attention of researchers across multiple industries, encompassing fields such as healthcare and manufacturing. Although blockchain's immutability, transparency, and traceability are undeniable advantages, persistent challenges limit its widespread integration. A significant problem encountered in blockchains is storage, particularly due to the growing number of blocks in the network. As the blockchain's size increases, it becomes harder for network peers to maintain pace, causing delays in transaction processing and reduced scalability. Furthermore, a larger blockchain requires more time to synchronize with new network nodes, which poses a significant challenge for new nodes who must download and verify the complete blockchain. This can create a bottleneck that restricts network scalability. Additionally, peers might encounter difficulties storing the entire blockchain, impeding their participation in network activities and transaction validation. Consequently, this situation could lead to centralization, where only peers with substantial storage capacities can engage in mining. Blockchain technology is undergoing rapid expansion, so innovative blockchain designs are needed. This article focuses on addressing the storage issue in current blockchain systems and exploring its impact on scalability. It explores current approaches for improving the storage efficiency of blockchain technology. These suggested solutions are grouped according to their methods, and their effectiveness is assessed through a comprehensive comparative analysis.

Keywords
Blockchain, Storage optimization, Network centralization, Transaction processing, Scalability.

1.Introduction
In an increasingly digital and interconnected world, the need for secure, trustworthy, and transparent systems is paramount. Originally associated primarily with cryptocurrencies like Bitcoin [1], blockchain has grown into a disruptive technology that has expanded its reach into sectors beyond finance. Blockchain, with its decentralized and unchangeable ledger, presents an innovative solution for addressing trust and security challenges, driving innovation in numerous domains. Fundamentally, a blockchain is a digital ledger that is distributed and decentralized, responsible for recording transactions throughout a network of computers. Each transaction is bundled into a block and linked in chronological order, forming an unbroken chain of blocks [2]. This decentralization ensures that no single entity controls the entire system, enhancing security and reducing vulnerabilities.

Blockchain relies on cryptographic techniques to secure data and enable secure transactions. Public and private keys ensure secure access and digital signatures validate transactions without the need for intermediaries. This cryptographic foundation safeguards data from tampering and reinforces the integrity of the entire system [3].

Blockchain technology has undergone a remarkable evolution across diverse applications since its inception [4]. Originally introduced as the foundation of cryptocurrencies like Bitcoin [1], blockchain has evolved into a multifaceted technology ecosystem. It's early emergence in cryptocurrencies set the path for the development of decentralized applications (dApps) and smart contracts, enabling the creation of programmable digital assets and self-executing agreements [5]. Over time, industries such as supply chain management adopted blockchain to enhance transparency, traceability, and authenticity verification of products [6]. The healthcare sector

*Author for correspondence

accepted blockchain for secure and interoperable medical records, while digital identity solutions emerged, offering users greater control over their personal information [7]. Voting systems found renewed trust through blockchain's tamper-resistant nature, ensuring transparent and auditable elections [8]. Financial services experienced disruption as blockchain ventured into cross-border payments, securities trading, and streamlined settlement processes [9]. Real estate transactions were simplified by tamper-proof property ownership records, while the energy sector explored blockchain for efficient energy trading and grid management [10]. Internet of things (IoT) devices gained enhanced security and interoperability through blockchain, and the art world saw the emergence of digital provenance and ownership verification [11].

1.1Current challenges

Despite its promising growth prospects, blockchain encounters challenges in areas like scalability, security, adoption, regulation, and interoperability [12-15].

(a)Scalability: Scalability of blockchain networks is the ability of that platform to support increasing load of transactions, as well as increasing the number of nodes in the network.

(b)Security: While blockchain is generally considered to be secure, there have been instances of hacks and attacks on cryptocurrency exchanges and other blockchain-based systems.

(c)Adoption: Blockchain technology is still in its early stages, and there is a lack of awareness and understanding among the public, which can hinder adoption.

(d)Regulation: There is currently a lack of regulatory clarity around blockchain and cryptocurrency, which can make it difficult for businesses and investors to operate in this space.

(e)Interoperability: Different blockchain networks can have different protocols and standards, which can make it difficult for them to communicate and work together seamlessly.

Blockchain network scalability is defined by its ability to effectively manage an expanding transaction volume and accommodate a growing number of network nodes. The primary challenge to achieving scalability in blockchain lies in the substantial data storage demands [16-21]. Since blockchain operates on an append-only basis, the data within the network continually expands. The continuous growth of data places a burden on the storage capacity of network participants, resulting in decreased system performance and limited transaction processing capabilities, which, in turn, impedes the broad adoption of the technology. The rise in Bitcoin transactions has resulted in several noticeable consequences. The average confirmation time for transactions has increased, reaching a point where there were 200,000 unconfirmed transactions and confirmation times exceeding a day [22, 23]. Additionally, the network transaction fee has risen significantly, reaching as high as $60 per transaction. The block difficulty has also increased, resulting in higher computational power consumption during block mining, consequently leading to increased electricity usage. Moreover, the size of the blockchain has expanded considerably, posing challenges for the setup of new full nodes responsible for maintaining the complete blockchain state for transaction processing and verification.

The design of Ethereum's blockchain aimed to offer greater flexibility compared to Bitcoin's, enabling developers to build dApps on the Ethereum network. However, as the number of dApps and network participants increased, the blockchain size experienced rapid growth, imposing significant storage demands on nodes [24]. As a result, the count of transactions in a pending state increased. As the blockchain grows, the increased data storage requirements can hinder node synchronization, increase costs, and potentially lead to centralization. Storage optimization in blockchain is crucial to manage the ever-growing data size, reduce storage costs, and ensure efficient node synchronization. It helps maintain decentralization and accessibility, enabling broader adoption of blockchain technology while addressing scalability challenges.

Various factors that affect scalability are latency, number of nodes, block size, computational cost, transaction cost, and storage or a high volume of data [16-20]. Addressing the scalability challenge requires ongoing research and innovation.

1.2Objectives

The primary factor limiting scalability is the storage capacity of peers in the blockchain network. Since Blockchain is append-only, the amount of data in the network continuously grows, leading to upward pressure on the storage space. The study aimed to conduct a systematic and methodical literature review on different approaches to optimizing storage in a blockchain network with the ultimate goal of enhancing scalability. Through meticulous management and optimization of storage needs, the

goal is to enhance scalability, enabling the network to accommodate a rising volume of transactions and participants without incurring unnecessary storage burdens. The primary highlights of this review encompass:

- Recognizing the requirement for enhancing storage efficiency approaches in blockchain networks.
- Developing a categorization framework for diverse storage optimization strategies in blockchain systems.
- Exploring the strengths and weaknesses of various techniques within each methodology.

1.2.1Research questions

This study aimed to address the objectives by reviewing the research questions formulated as follows:

RQ1: How to address blockchain storage challenges for ledger growth efficiency?

RQ2: What are the criteria for efficient storage categorization in blockchain optimization framework?

RQ3: How do storage optimizations impact blockchain scalability and performance metrics?

1.2.2Systematic review process

The preferred reporting items for systematic review and meta-analysis (PRISMA) methodological approach conducted for the study includes three stages identification, screening, and inclusion, as represented in the flowchart in *Figure 1*, and a detailed discussion can be found in section 2 of the study. Section 3 introduced a framework for categorizing various storage optimization strategies in blockchain systems, and section 4 offered an analysis of techniques employed to enhance the storage efficiency in the blockchain.

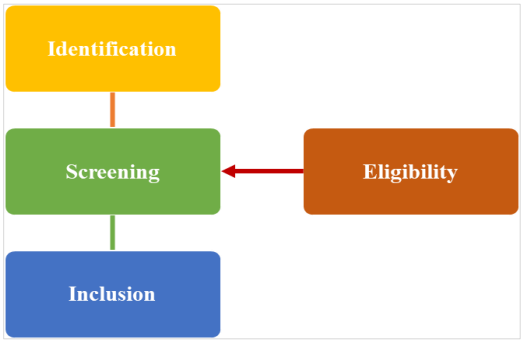

Figure 1 Stages of PRISMA methodology

2.Review methodology

This paper primarily focuses on investigating methods for optimizing storage in blockchain technology. The comprehensive investigation in this study involved a systematic literature review using the PRISMA methodological approach, which encompasses three primary stages, along with the incorporation of inclusion and exclusion criteria. The PRISMA representation for the review and analysis is shown in *Figure 2.*

2.1Stage1: identification

Various digital databases, including IEEE Xplore, ScienceDirect, Springer, ResearchGate, MDPI, Academia, and others, were analysed to find research papers focused on optimizing storage in blockchain technology. The search phrases such as "optimizing storage in blockchain technology", "storage optimization in blockchain" was used, and a multi-step process was employed to filter and identify relevant papers. Initially, papers were screened by examining their titles, abstracts, and conclusions. Following this preliminary assessment, the selected papers underwent a thorough reading.

From the search results, Google Scholar yielded 854 research articles published between 2016 and 2023. IEEE Xplore retrieved 191 papers released between 2018 and 2023, while SpringerLink returned 1,799 research papers.

2.2Stage2: screening

In screening stage articles are involved based publication date between 2016 and 2023, which are focuses on addressing storage and scalability issues in blockchain systems, and having detailed descriptions of the proposed methods. The articles which do not focus on storage and scalability challenges within blockchain systems are excluded for review. *Table 1* outlines the criteria used to determine which papers to include in our review whereas and criteria we used to exclude the papers to include in our review.

Table 1 Inclusion and exclusion criteria for selected work

S. No.	Criteria	
1	**Inclusion criteria**	Publication date between 2016 and 2023 Focus on addressing storage and scalability issues in blockchain systems Detailed descriptions of the proposed methods
2	**Exclusion criteria**	Articles that do not focus on addressing storage and scalability challenges within blockchain systems.

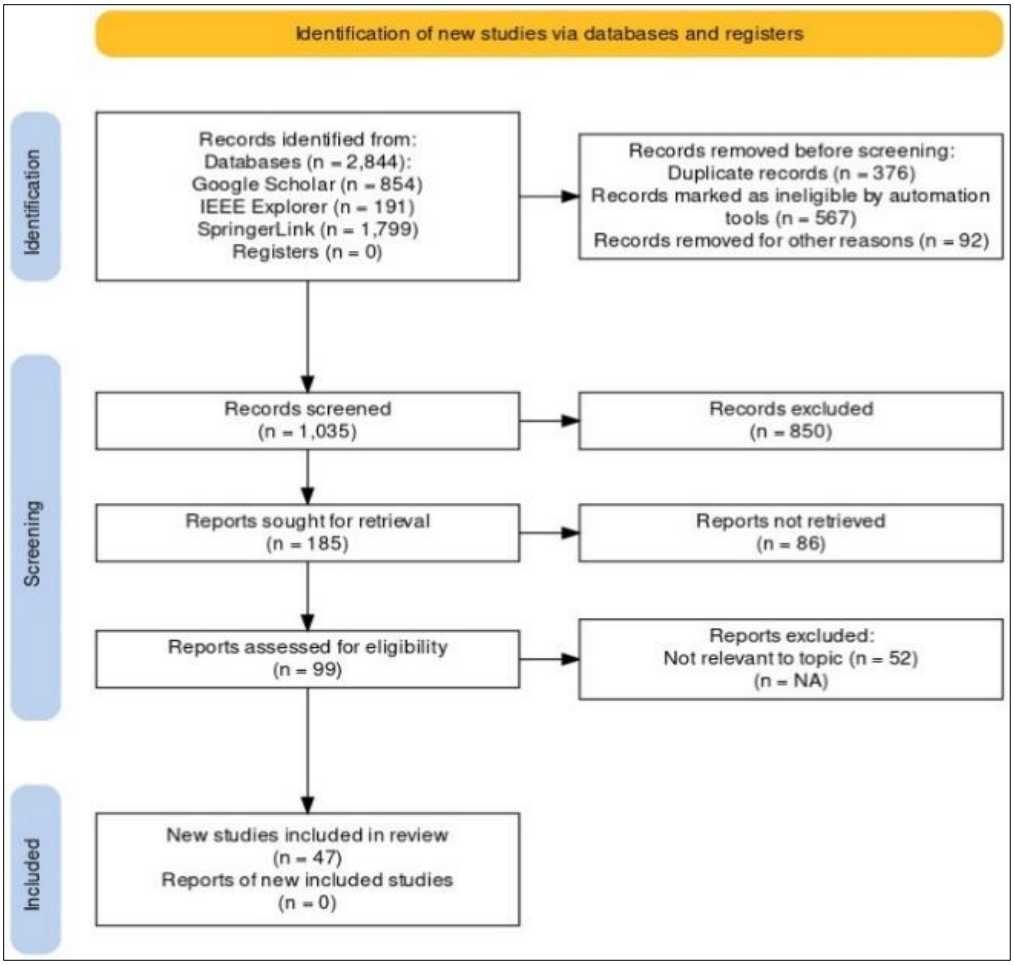

Figure 2 The PRISMA for review and analysis

2.3Stage3: inclusion

The process included the careful selection of articles that closely aligned with the research's aim, scope, and objectives. These chosen articles were thoroughly examined to extract relevant information for the intended research, resulting in the development of a literature review comprising 46 articles. This review aimed to identify additional research, significant findings, and conclusions. *Table 2* presents a year-wise breakdown of papers related to techniques for optimizing blockchain storage. It indicates that a greater number of journal papers have been considered in comparison to conference papers, with the majority of these papers dating from the years 2016 to 2023. Additionally, the table illustrates the yearly percentage of paper usage. *Table 3* shows the summary of number of research papers included from various publishers.

Numerous methods have been investigated to address the storage challenge within the domain of blockchain by examining of 46 included papers. Numerous studies have proposed remedies aimed at mitigating scalability challenges, often introducing enhancements or models to optimize storage. These solutions are typically tailored to specific use cases and are applicable in both permissioned and permissionless blockchain environments. Among the techniques used to handle and store data more efficiently within a blockchain network, two prominent strategies are on-ledger and off-ledger storage optimization. On-ledger storage optimization involves the optimization and direct storage of data within the blockchain itself. In contrast, off-ledger optimization requires the storage of data outside of the primary blockchain [24]. *Figure 3* provides a comprehensive understanding of these storage optimization methods. *Figure 4* illustrates the distribution of research within the on-ledger and off-ledger storage optimization categories, indicating that a minimal amount of research has been dedicated to off-ledger storage optimization.

Table 2 Year-wise count of papers and percentage

S. No.	Year	Journal articles	Conference articles	Total	%
1	2016	NA	1	1	2%
2	2017	NA	1	1	2%
3	2018	1	7	8	17%
4	2019	5	4	9	20%
5	2020	9	1	10	22%
6	2021	10	NA	10	22%
7	2022	6	NA	6	13%
8	2023	1	NA	1	2%
Total		**33**	**14**	**47**	**NA**

Table 3 Summary of number of research paper included from various publisher

S. No.	Publisher	No. Of research paper includes
1	IEEE	31
2	Elsevier	4
3	ACM	4
4	Springer	3
5	Wiley	1
6	Oxford University Press	1
7	other	1

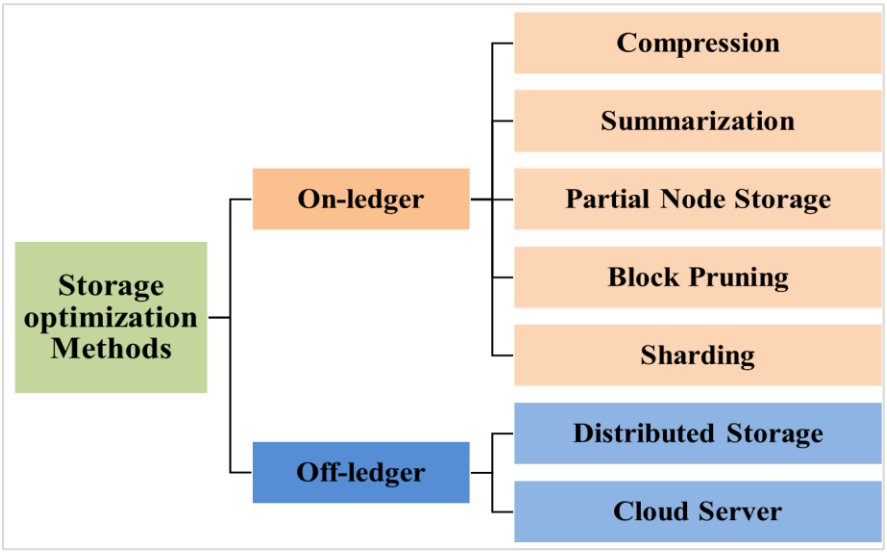

Figure 3 Categorization for storage optimization strategies of the blockchain network

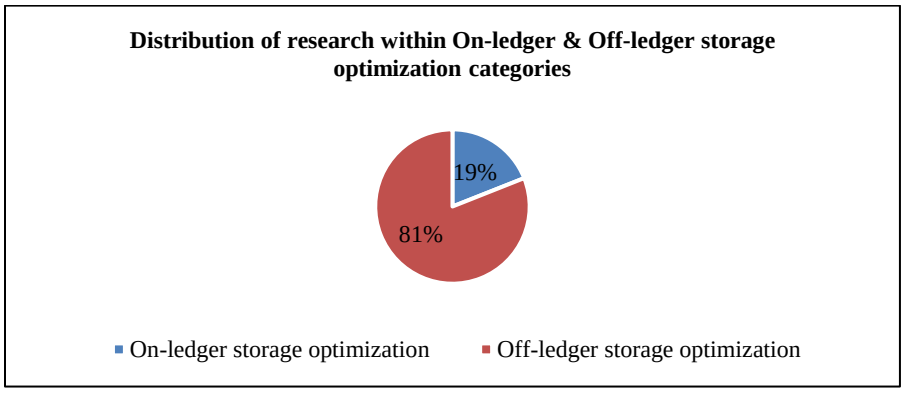

Figure 4 Percentage of research distribution within on-ledger and off-ledger storage optimization

3.Literature review

This section, explores various solutions are explored to address the challenge of storage burdens in blockchain systems. The categorization introduced in this research is based on the techniques used to attain storage enhancement.

3.1On-ledger storage optimization

On-ledger storage optimization involves the procedure of diminishing the data volume stored within blockchain blocks. To accomplish optimized storage, diverse methods are employed, including compression, summarization, partial node storage, block pruning, and sharding. The graphical representation in *Figure 5* showcases the distribution of research articles focused on on-ledger storage optimization, relative to the total number of research articles reviewed in each corresponding year. Additionally, it provides insights into the growth of research and literature trends in this domain over time.

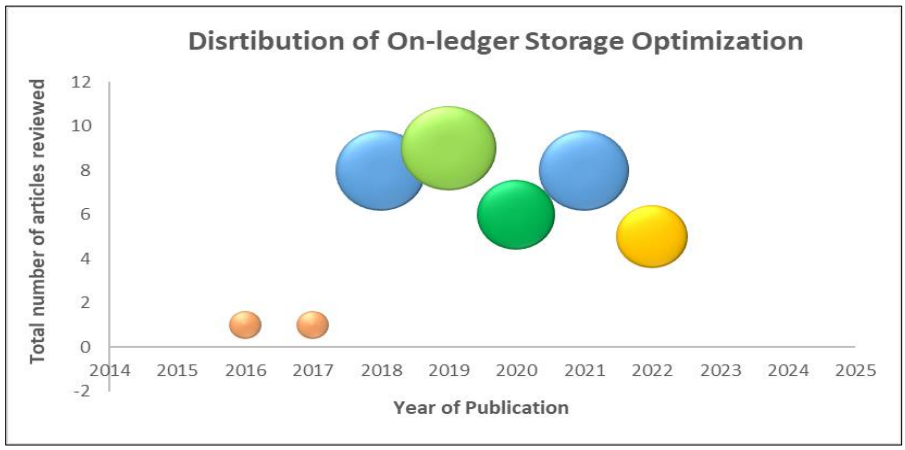

Figure 5 Distribution of on-ledger storage optimization

3.1.1Compression

In a blockchain network, compression is the practice of employing specialized algorithms to shrink the size of stored data. This reduction in data size conserves valuable storage space, critical in a system where data accumulates continually. This efficiency boost results in faster transaction processing and propagation, contributing to a more responsive and scalable blockchain network. In essence, compression helps strike a balance between data preservation and network optimization. *Table 4* provides an overview of the compression method used to enhance storage efficiency.

Kim et al. [25] introduced, a solution named the selective compression (SELCOM) scheme. This scheme utilizes a Block Merkle Tree (BMT) and aims to address the storage challenges faced by blockchain nodes with constrained resources. The SELCOM scheme facilitates effective management of storage capacity by enabling nodes to choose which blocks to retain. This selection is managed through a proposed checkpoint chain, effectively forming a secondary blockchain. SELCOM is conducted to minimize the storage capacity required for the blockchain, retaining only essential data on each node for verification purposes. The framework comprises four key procedures: compression, checkpointing, updating, and selection. In the compression phase, recently gathered blocks undergo compression to create a fresh BMT. This BMT then forms the basis for generating a checkpoint during the checkpointing process. As checkpoints accumulate, the update process is executed to streamline them and alleviate the buildup of numerous checkpoints. Finally, in the selection stage, each node individually determines which blocks to retain for verification purposes, subsequently removing the remaining blocks to effectively minimize storage requirements. The suggested approach retains compressed results within a secondary blockchain, necessitating communication to accomplish live synchronization among nodes.

Yu et al. [26] presented a dual approach to optimize transactions and compress blocks, with the primary goal of reducing the overall block size in the blockchain. The transaction optimization model achieves this objective by reducing the size of individual transactions, focusing on both Coinbase and regular transactions. Moreover, the block compression model employs a potent data compression algorithm to further shrink the size of blocks. In the Bitcoin blockchain, each block consists of several

components, including block size, block header, and a transaction counter, all of which have limited room for significant reduction. Conversely, the volume of transactions within each block is substantial, containing extensive data, some of which may be redundant or retrievable from the blockchain. Through the transaction optimization model, the transactional data structure undergoes refinement, leading to a decrease in the overall transaction data size. Simultaneously, the block compression model leverages an efficient data compression algorithm to achieve a reduction in block size.

Chen et al. [27] introduced a technique aimed at reducing the storage requirements of the Bitcoin blockchain. In a Bitcoin block, there are two main components: the block's header and its body. The block's header, occupying 80 bytes, includes crucial block details like the version number and the hash of the preceding block. Their approach involves replacing the previous block's hash with index pointers. The method they proposed consists of two primary phases: compression and decompression. During the compression phase, for each transaction within the Bitcoin block, the previous transaction hash field is identified and replaced with an index. Similarly, in the decompression process, a method is employed to restore the previous transaction hash using the provided index.

Qi et al. [28] presented a framework called compressed and private data sharing (Cpds). Cpds employs compression and encryption to secure product data before it is placed onto the blockchain. First, Cpds uses a tree-based compression mechanism to allow industrial stakeholders to compress their product data efficiently. Tree-based compression mechanism divides the product data into a hierarchical tree structure and then compresses each node in the tree. This allows Cpds to compress the product data efficiently without losing too much information. Second, Cpds employs a hybrid access control strategy to encrypt product data, ensuring that authorized individuals, including both industry participants and third-party users, can gain access to the encryption keys. The hybrid access control approach uses a combination of public key cryptography and secret sharing to encrypt the product data. This ensures that only authorized users can access the data, even if some of the secret keys are compromised. The Cpds framework addresses the storage challenge of sharing product data on the blockchain by overcoming challenges such as privacy and Access control.

Guo et al. [29] proposed residue number system based adaptive compression (R-ABC) scheme to address the storage limitations of blockchain networks. This scheme compresses the block body by expressing the transactions as their corresponding remainders and distributing them across the network. The original transactions can be reconstructed using the Chinese remainder theorem (CRT). Additionally, a regularization technique is introduced to handle transactions of different sizes, but it takes more time to implement. In the R-ABC scheme, the transactions in the block body are expressed as their corresponding remainders modulo a predetermined number. The remainder are distributed across the network. The original transactions can be reconstructed using the CRT. A regularization technique is used to handle transactions of different sizes. This technique involves padding the smaller transactions with zeros. The R-ABC scheme is a promising approach to addressing the storage limitations of blockchain networks. It is simple to implement and can achieve significant compression ratios.

Marsalek et al. [30] put forth a novel blockchain architecture with built-in compressibility features, aimed at reducing the overall size of the blockchain. Their proposal centered around a snapshot-oriented strategy, where periodic snapshot blocks are created. These snapshot blocks encompass the entire unspent transaction output (UTXO) dataset and block headers. These snapshots collectively form a secondary snapshot chain, which can be stored efficiently on devices with limited resources or constraints. This design is particularly well-suited for blockchains that adhere to the UTXO model, as seen in Bitcoin, for instance. However, it's important to note that this approach introduces a significant level of complexity because maintaining this secondary chain requires synchronization among network nodes. It's worth mentioning that the framework doesn't address the issue of the cumulative impact of accumulating snapshots over time. To improve this concept, it would be beneficial to introduce mechanisms for managing and reducing the long-term accumulation of these snapshots.

Table 4 Summary of compression approach for blockchain storage optimization

Authors	Method	% of Storage Reduction	Weakness
Kim et al. [25]	SELCOM utilizes BMT	76.02%Reduction in storage space	The secondary blockchain needs real-time node syncing
Yu et al. [26]	The transaction optimization model and block compression model	34.81%, Compression Ration	Higher mining complexity & loss of data
Chen et al. [27]	Replacement of the previous block's hash with index pointers	Reducing Bitcoin blockchain storage by up to 12.71%	Low storage reduction ratio.
Qi et al. [28]	Cpds uses a tree-based compression	4-9 times	As data size increases, the compression ratio decreases
Guo et al. [29]	R-ABC scheme & and CRT for decompression	80% Reduction in storage space	More time consumption
Marsalek et al. [30]	Snapshots are taken periodically, forming a linked second blockchain.	75% Reduction in storage space	Complexity because of maintaining this secondary chain. & accumulation of snapshots over time

3.1.2Summarization

Block summarization in a blockchain network refers to the process of condensing or summarizing the information contained within a block, typically to optimize storage, enhance data retrieval efficiency, and improve the overall scalability of the blockchain. Block summarization addresses this challenge by creating compact representations. *Table 5* provides a summary of the approach to optimize storage effectively through summarization.

Xu et al. [31] introduced an approach called efficient public blockchain client (EPBC), which addresses the challenge of enabling resource-constrained users, such as IoT devices and smartphones, to actively participate in blockchain applications without the necessity of storing the entire blockchain. The core idea behind this approach revolves around the summarization of the blockchain into a compact, fixed-size summary. This summary serves as the sole data requirement for lightweight users denoting individuals or devices with limited storage and computational capabilities. For these users, there's no need to maintain or retrieve the complete blockchain, which can be exceptionally voluminous. Instead, they have to allocate storage space for summary, and this allocation remains relatively unaffected by the blockchain's overall size. In this approach, lightweight users retain the capacity to validate the authenticity of blocks and transactions, all the while significantly reducing the storage and resource burden for those operating with constrained devices. EPBC consists of four algorithms (setup block, summary construction, proof generation, and proof verification).

Dorri et al. [32] introduced a blockchain framework known as the memory optimized and flexible blockchain (MOFBC). This framework creatively addresses issues related to transaction removal and reducing the size of data within the context of blockchain. Typically, the immutability of blockchain prevents the removal of data from the ledger. However, MOFBC overcomes this limitation by introducing a mechanism that allows for the elimination of previously recorded transactions or the condensation of their size through a process called transaction summarization or data aging. In this approach, MOFBC consolidates multiple transactions into a single summary, improving the efficiency of data management. Unlike conventional methods that hash the entire content of transactions, MOFBC hashes the transaction hashes themselves. This approach enables the removal of transaction content while preserving the hash within the blockchain, ensuring data integrity is maintained. To achieve this summarization, MOFBC introduces the concept of a summary transaction, which facilitates the consolidation of multiple transactions into one comprehensive entry. This approach offers improved memory optimization and flexibility when compared to traditional blockchain models, effectively addressing challenges related to transaction removal and an overall reduction in the size of the blockchain. The processing time for transactions that can be summarized is longer than for temporary transactions.

Nadiya et al. [33] devised an approach aimed at streamlining the verification of payments within a blockchain network by introducing a method for summarizing and compressing blocks. This method not only simplifies the tasks performed by network nodes but also offers substantial benefits through the creation of compressed summary blocks characterized by their reduced data size. The essence of these summary blocks lies in their ability to encapsulate hash references to the original blocks, thereby preserving the crucial linkages required for validating

future transactions associated with the summarization block. To achieve this, a multi-step process is employed. Initially, the formed summary block captures essential data from the original blocks. Following this step, the summary block undergoes compression using the deflate compression algorithm. The application of the deflate compression algorithm is pivotal in achieving space efficiency. This compression process is instrumental in optimizing the storage of blockchain data by reducing the volume of information within these summary blocks.

Palai et al. [34] introduced an innovative approach designed to summarize blocks that undergo modifications specifically customized for lightweight nodes. The primary objective of this methodology is to maintain the operational efficiency of thin clients while affording them certain capabilities typically associated with full nodes. The block summarization process, at its core, requires the substitution of actual blocks with corresponding summary blocks. These summary blocks encapsulate essential information, such as details about spent transactions and unspent outputs, meticulously extracted from the original block summary. This strategic adaptation equips thin clients with the capacity to authenticate incoming transactions effectively. Furthermore, in scenarios where nodes possess substantial computational resources and processing power, they may potentially engage in the mining process for subsequent blocks. This groundbreaking advancement serves to significantly reduce the dependency of lightweight nodes on their network peers, marking a notable stride in the evolution of blockchain technology.

Shi et al. [35] introduced reliable and efficient storage scheme (RESS), a node storage scheme, which uses Raptor codes to encode blocks. In the Bitcoin network, each node stores only some of these coded blocks. RESS employs a block grouping strategy, encoding and storing groups of blocks regularly. This allows for efficient verification of transactions in new blocks, reducing the need for frequent data decoding by nodes.

Table 5 Summary of summarization approach for blockchain storage optimization

Authors	Method	% of Storage Reduction	Weakness
Xu et al. [31]	EPBC which uses blockchain Summarization	---	EPBC makes heavyweight clients with local ledgers run separate EPBC instances for associated tasks.
Dorri et al. [32]	MOFBC consolidates multiple transactions into a single summary	Reduction in memory usage by 25%	The time required for processing summarizable transactions is more
Nadiya et al. [33]	Summarization algorithm and deflate compression algorithm	The reduction in space for summary blocks is 22.318%, while for compressed summary blocks, it is 78.104%.	Created for the Bitcoin blockchain, there is no established standard for summary blocks in other blockchain systems
Palai et al. [34]	Block Summarization	50% to 60% compression ratio.	As the number of transactions increases, fragmentation diminishes & a higher proportion of light nodes results in reduced availability of original blocks.
Shi et al. [35]	RESS uses Raptor codes, and block grouping for efficient Bitcoin storage.	89%	Decoding requires additional time.

3.1.3Partial node storage

Partial no storage in blockchain refers to a modified node configuration where the node stores only a portion of the entire blockchain's transaction history, rather than the complete history. This approach aims to reduce storage requirements while still enabling the node to participate in transaction validation and consensus processes. It's a compromise between full nodes, which store the entire history and require significant resources, and light nodes, which rely on others for transaction information. The concept seeks to strike a balance between resource efficiency and network participation. *Table 6* shows a summary of the storage optimization approach using partial node storage.

Qi et al. [36] have introduced a storage engine named BFT-Store, designed to enhance storage scalability within permissioned blockchain environments. The BFT-Store is the combination of erasure coding with the byzantine fault tolerance (BFT) consensus protocol, offering a solution for the limitations inherent in the conventional full-replication storage approach. Erasure coding, a fundamental component

of BFT-Store, operates by transforming initial blocks into a sequence of encoded blocks referred to as "chunks." These chunks possess a unique property: the ability to reconstruct the original data from any subset containing a sufficient number of available chunks. To ensure resilience and distribution, these encoded chunks are disseminated across all nodes in the network, totaling 3f+1 nodes, where 'f' denotes the quantity of potentially faulty nodes. In the context of the practical application of the practical BFT (pBFT) consensus protocol, a minimum of 2f+1 nodes are equipped with the necessary chunks to successfully reconstruct the original data.

Liu et al. [37] have proposed a streamlined blockchain framework known as LightChain, which incorporates a novel feature called the unrelated block offloading filter (UBOF). UBOF operates by analyzing the UTXO set within the blockchain. Its primary objective is to identify and classify unrelated blocks (UBs). These UBs are characterized by the condition where all of their transaction outputs(TXO) have been fully spent and are no longer directly referable in future transactions. UBOF excels in identifying and subsequently removing these UBs from the blockchain, a process that leads to a significant reduction in the overall storage footprint occupied by the blockchain.

Li et al. [38] have introduced a strategy for optimizing blockchain storage, centered around the use of Reed-Solomon (RS) erasure code. In this scheme, upon the creation of a new block within the blockchain, it undergoes a process of segmentation using the RS erasure code. Subsequently, an appropriate selection of nodes is made to store these segmented code blocks. During the block recovery phase, a designated recovery node can request the required coded segments from other nodes until a sufficient number of coded segments are obtained for the decoding process. Once this criterion is satisfied, the original block can be successfully reconstructed. Within the blockchain network, full nodes maintain comprehensive structural and transactional data for each block. As a result, full nodes are capable of providing data validation services to other nodes and can synchronize the entire blockchain. In contrast, light nodes store the segmented code blocks. If any part of the coded block is tampered with by malicious actors, the root hash value of the reconstructed block will be altered, allowing for a comparison with the hash value of the corresponding block stored on a full node.

Chen et al. [39] have presented a novel data organization concept suitable for consortium blockchains, which integrates both on-ledger and off-ledger data management. This approach prioritizes the retention of essential information within the blockchain while relegating less critical, extensive data to an external centralized database. "It involves the procedure of partitioning the data and saving it in both blockchain and a central database. The majority of the data can be categorized into two segments: the 'core' and 'non-core' components. This strategic partitioning serves to significantly reduce the overall data footprint within the blockchain network. The methodology at the core of this approach entails the inclusion of a hash generated from the off-ledger data within the blockchain block. Subsequently, this hash is meticulously monitored and linked using a structured Merkle tree arrangement, facilitating data verification and traceability processes.

Mei et al. [40] have introduced a storage optimization technique centered on the adoption of a residual number system (RNS). The primary aim of this approach is to minimize the storage requirements imposed on individual nodes in a blockchain network. To fortify the robustness of their storage methodology, the researchers have integrated the data recovery process based on CRT-II (an advanced variant of the CRT). This addition serves to identify corrupted data within nodes that may exhibit malicious behaviour, significantly enhancing the resilience of the proposed storage strategy against potential faults. The fundamental concept underlying this strategy entails the storage of only the remainder of account information, utilizing a significantly smaller modulo. By adopting this approach, the storage demands placed on each node are substantially reduced. Within Mei and colleagues' framework, a predefined set of modulos is established, and upon joining the network, each node selects one of these modulos to use in its operations.

Zhao et al. [41] introduced an innovative lightweight node known as enhanced simplified payment verification (ESPV). This lightweight node adopts a unique approach by retaining full copies of the most recently generated blocks, enabling it to execute transaction verification. As for older blocks those generated before the latest block, they are partitioned into smaller segments to minimize data duplication. These older blocks are sliced and stored with a focus on upholding the reliability and accessibility of blockchain data, thereby reducing storage space wastage and improving the system's scalability.

Furthermore, the entire block header information of the blockchain is retained to ensure the genuineness of the blockchain data in the system.

Xu et al. [42] presented the idea of a consensus unit (CU), which consolidates multiple nodes into a unified entity, allowing them to jointly maintain at least one copy of the blockchain data within the system. Not all participants are equally concerned with the entire state of the system; some may only be interested in specific transaction segments. The blockchain data blocks are broken into pieces and distributed among different nodes. Each participant stores complete block headers, which enable them to utilize the Merkle tree root within each block header to verify the block data they receive from other sources.

Xu et al. [43] presented a section blockchain model, where nodes are required to store specific components such as a block header, a designated set of blockchain fragments which are groups of blocks, and a predefined number of database snapshots (representing the status of records, such as account balances at specific points in time). Miners, on the other hand, must include a specific quantity of fragments and database snapshots when they engage in the mining process. To be eligible for participation in the mining process, miners must meet a minimum requirement for both fragments and database snapshots.

Li et al. [44] presented the grouping storage scheme (GAPG), employing the CU grouping concept. This scheme assembles various nodes into a single cluster and assigns all the blocks from the entire chain to each cluster of nodes. Every node within a cluster is required to retain all the block headers, as the Merkle root hash can be used to validate the legitimacy of transaction data. In this study, an algorithm named generalized assignment problem is employed to distribute blocks to nodes within each cluster based on a heuristic approach.

Liu et al. [45] introduced a security scheme involving two interconnected chains to address the issue of storage limitations in certain nodes. In this system, there are three types of nodes. First is demanding nodes (DNs), the nodes require additional storage space and seek to borrow it from other nodes. Second is providing nodes (PNs), the nodes possess sufficient storage capacity and offer storage services to other nodes in need. Third is computing nodes (CNs) which are not DNs or PNs categories but play a role in resource allocation calculations. The first chain, known as the information chain, primarily stores user information such as IDs, internet protocol (IP) addresses, available disk space, total storage size, and similar data. The second chain, known as the transaction chain, serves as the primary ledger for documenting transactions involving DNs and PNs. When DNs initiate storage requests, a randomly chosen CN accesses the most up-to-date data from the information chain to determine how to allocate memory. The outcomes are subsequently conveyed back to the relevant DNs and PN. The memory allocation algorithms employed in this system include the fast-matching algorithm (FMA), a greedy approach, the genetic algorithm (GA), and the tabu search algorithm (TSA).

Dai et al. [46] presented a data reduction method called jigsaw-like data reduction approach (Jidar), which operates like a jigsaw puzzle. In the Jidar approach, each node selectively retains specific transactions of relevance and the corresponding Merkle branches from complete blocks, like handpicking particular pieces from a jigsaw puzzle. For instance, let's consider node A with 4 transactions namely transaction-0, transaction-1, transaction-2, and transaction-3 as an example. As transaction-3 is associated with node A, the node only stores transaction-3 along with the Merkle branch that connects transaction-3 to the root of the tree. It discards the other branches, including transaction-0, transaction-1, and transaction-2, as they are not pertinent to its interests.

Table 6 Summary of partial node storage approach for blockchain storage optimization

Authors	Method	% of Storage reduction	Weakness
Qi et al. [36]	BFT-Store (Integrating erasure coding with BFT) which stores encoded chunks of blocks	86%.	Takes more time to decode
Liu et al. [37]	UBOF which identifies and removes UBs	43.35%	Historical Data Loss
Li et al. [38]	RS erasure code where blocks are segmented and stored on different nodes	46.78%	Takes more time to reconstruct of original block
Chen et al. [39]	Divides data into parts, then stored in blockchain and central database	--	Less secure as some data is stored in the central database & a high query processing time

Authors	Method	% of Storage reduction	Weakness
Mei et al. [40]	Storing only the remainder of account information using RNS and CRT-II to recovery the complete account information	90%	Fault (devil nodes) detection is not done
Zhao et al. [41]	ESPV retains full copies of the most recently generated blocks. Older blocks sliced and stored on nodes	75%	The UTXO support for older blocks is lacking
Xu et al. [42]	CU in which blocks are broken into pieces and distributed among different nodes	75%–95%	Significant delays when performing queries of the local node
Xu et al. [43]	Section blockchain: nodes store block header, fragments, snapshots	A node only requires 0.2% of the total storage.	Data loss
Li et al. [44]	GAPG Grouping Storage Scheme, employing the CU grouping concept where Blocks are assigned to nodes, duplicates fill the remaining space.	---	As block count rises, fixed node capacity exhausts storage space. CU scheme helps, but no extra room for duplicates.
Liu et al. [45]	FMA involves two interconnected chains to address the issue of storage limitations in certain nodes	--	Only suitable for a small number of nodes
Dai et al. [46]	Jidar in which Nodes keep relevant transactions and their Merkle branches for the Bitcoin system	Decrease a typical node's storage expense to approximately 1.03%.	Increases the cost

3.1.4Block pruning

Block pruning refers to a process of removing certain data from the blockchain's history. Pruned nodes can discard certain data that is no longer needed for validation, which can significantly decrease the amount of storage space required to participate in the network. Pruned nodes achieve this by keeping a subset of the blockchain data, including the most recent blocks and a portion of the transaction history. *Table 7* presents a summary of the optimization of storage through the technique of block pruning.

Wang et al. [47] introduced an effective storage strategy (ESS), which takes advantage of the distribution patterns of UTXOs within the Bitcoin network. ESS assigns a weight to each UTXO and utilizes this weight to selectively trim blocks that are queried less frequently. In the ESS node, recently created blocks with a greater quantity of UTXOs retain full block details, while other blocks merely retain the block headers. This setup enables the independent verification of most transactions. However, for a subset of older blocks that contain UTXOs, when they are utilized as inputs for new transactions, a request is sent to a full node for verification.

Matzutt et al. [48] introduced a pruning scheme based on snapshots that is entirely compatible with Bitcoin. In this method, when new nodes join the network, they only need to obtain small snapshots to bootstrap, allowing them to remove outdated data without negatively affecting the overall network's health. Nodes that adopt this method, known as CoinPrune nodes, regularly create snapshots of their UTXO collection. Instead of transmitting the entire blockchain history, they share these snapshots with new nodes, leading to decreased requirements for storage, bandwidth, and processing power. These snapshots are synchronized with the current block height, signifying the location of the latest block in the blockchain, and they encompass an ordered UTXO set intended for synchronization and verification functions.

Heo et al. [49] presented a strategy called multi-level distributed caching (MLDC) designed to enhance blockchain storage efficiency by minimizing data duplication in line with decentralized data access patterns. MLDC incorporates a hierarchical storage classification (SC) system in which each node is assigned to a specific SC, equipped with its unique access frequency (AF) threshold, which is determined based on the node's availability. To reduce the redundancy of shared data among participant nodes, nodes within an SC periodically eliminate data that hasn't been accessed for a specific duration as dictated by the AF threshold for that particular SC. Nevertheless, they retain all block hashes for the sake of consistency. Over time, all nodes within the MLDC network store the most frequently accessed data in their local storage, thus promoting efficient data management.

Pyoung et al. [50] proposed LiTiChain, which is a blockchain consisting of blocks that have a limited lifespan. In LiTiChain, transactions and blocks that

have become outdated, meaning their lifetimes have expired, can be securely eliminated from the blockchain. The lifespan of a block is determined as the duration from when the block is created to the most recent endpoint in time among transactions. The endpoint of a transaction or block is identified as the specific time index when its lifespan concludes. If a block's lifespan comes to an end, it can be removed from the blockchain to free up storage resources on edge servers and enhance security. This method combines two kinds of structures: the first structure forms a tree-like graph based on the sequence of lifespan expiration, while the second structure follows a linear pattern similar to the conventional blockchain, based on the order of creation.

Gao et al. [51] introduced a mechanism designed to periodically summarize blocks from the past, retrieve and arrange the UTXO data they contain, and remove unnecessary transaction information from the Bitcoin blockchain. This process occurs at defined intervals, where a designated summary block takes on the responsibility of systematically purging block data. Miner nodes gather transaction output write files and store them on the inter planetary file system (IPFS) network. The hash value required to retrieve these files is then added to the summary block's header. The process of mining and broadcasting these summary blocks mirrors that of standard blocks. Upon receiving a summary block, other miner nodes must download and validate the UTXO files from the IPFS network. If the validation is successful, the miner node goes on to confirm the transaction details within the summary block using the standard block validation procedure. Once the summary block is approved by the majority of nodes in the network and becomes part of the local blockchain, it marks the conclusion of processing TXO from the Genesis Block to the most recent block. In response, a node has the option to remove the block's main content from its locally stored blockchain, with minimal impact on miner nodes. The UTXO data required for future transaction verification by miner nodes is securely stored on the IPFS network and can be retrieved locally when needed during the validation of summary blocks. Therefore, the removal of historical blocks minimally affects miner nodes.

Chen et al. [52] presented MiniChain, a method that replaces the UTXO set with two unalterable data structures: the spent transaction outputs (STXO) set and the TXO set. To qualify as a valid UTXO, an output must belong to the TXO set while not appearing in the STXO set. MiniChain employs three primary strategies. Initially, instead of a UTXO commitment, MiniChain utilizes an Rivest–Shamir–Adleman (RSA) accumulator to generate an STXO commitment in each block. The STXO set contains all previously spent coins and acts as an append-only data structure, eliminating the need for resource-intensive deletion operations. Secondly, to prevent users from creating non-existent coins, additional proof of existence is required. This can be conveniently achieved by providing a Merkle proof indicating the coin's origin. However, this approach necessitates validators to store all block headers and access them quickly. As the number of blocks increases, the efficiency of retrieving these headers decreases. To address this challenge, the proposal introduces a TXO commitment using Merkle mountain range (MMR). This allows validators to store only the MMR peaks, which is a logarithmically sized set compared to the blockchain's length, ensuring efficient verification of the presence of any coin in the blockchain.

Table 7 Summary of block pruning approach for blockchain storage optimization

Authors	Method	% of Storage reduction	Weakness
Wang et al. [47]	ESS in Bitcoin, assigning UTXO weights to optimize storage by selectively retaining full block details for high-UTXO blocks and headers for others.	82.14%	Constrained by the UTXO-based architecture
Matzutt et al. [48]	CoinPrune where Nodes create UTXO snapshots, sharing them with new nodes to avoid transmitting the entire blockchain history.	86.98%	Constrained by the UTXO-based architecture
Heo et al. [49]	MLDC by minimizing data duplication, using hierarchical storage, and retaining block hashes for consistency	83%	Requires a central authority to assign nodes to storage classes.
Pyoung et al. [50]	LiTiChain, blocks that have become outdated can be securely eliminated from the blockchain	Average storage is about 100%–140% of the baseline storage	Data Loss

Gao et al. [51]	Summarize past blocks, organize UTXO data, and remove unnecessary transaction details in the Bitcoin blockchain	--	Deleted transactions are not recorded.
Chen et al. [52]	Minichain which replaces the UTXO set with STXO and TXO sets. Valid UTXOs are in TXO but not in STXO	--	MiniChain is not compatible with existing UTXO-based blockchains

3.1.5Sharding

Sharding aims to address this scalability issue by dividing the blockchain network into smaller partitions called "shards." Each shard is responsible for processing and storing a subset of the total transactions. This means that not every node needs to process every transaction, which can significantly increase the network's throughput and reduce processing times. But sharding faces cross-shard communication overhead. *Table 8* displays a summary of the approach involving sharding for the optimization of storage.

Mizrahi et al. [53] proposed the concept of state sharding, a sophisticated technique employed in blockchain technology. State sharding implies the partitioning of a blockchain's state into multiple distinct shards, each of which is entrusted to a specific subgroup of network nodes. Their paper introduced an approach to state sharding, focusing on the utilization of space-aware representations. A space-aware representation refers to a method of mapping data onto these shards with a primary goal of minimizing inter-shard communication. This is achieved by grouping those portions of the blockchain state that are frequently accessed by transactions, thereby reducing the need for cross shard data transfers. Furthermore, their work proposed various algorithms designed for the identification of space-aware representations and the efficient computation of the shard assignments for a given transaction.

Cai et al. [54] introduced the many-objective optimization algorithm that relies on the dynamic reward and penalty mechanism (MaOEA-DRP). This algorithm aims to improve the shard validation validity model by tackling a specific concern: the potential for shard invalidation due to the presence of malicious nodes in a single shard, especially when the total number of malicious nodes is fewer than one-third of all nodes participating in the sharding system. MaOEA-DRP combines GA and simulated annealing techniques to accomplish its objectives. However, it's worth noting that this approach primarily focuses on evaluating the potential aggregation of malicious nodes and does not take into consideration the potential implications of anonymous address clustering, particularly about the privacy and security concerns associated with sharding in the context of the industrial internet of things (IIoT).

Jia et al. [55] presented an enhanced data storage model, leveraging blockchain sharding technology for optimization. Within this model, each node employs a technique called extreme learning machine (ELM) to categorize blocks into two distinct groups: hot blocks and non-hot blocks. This categorization is based on factors such as block popularity, storage demands, and historical query records maintained by nodes. Following this classification process, each node retains the most relevant hot block. When a node initiates a query, it can perform local queries, thus reducing the need for frequent query requests sent to other nodes across the network. The primary objective here is to store the most commonly accessed blocks in the most easily accessible shards while simultaneously minimizing the storage requirements for each shard. This approach aims to optimize data retrieval and storage efficiency within the blockchain-sharding context.

Xu et al. [56] proposed the concept of a segmented blockchain, a novel approach that divides the blockchain into segments, allowing nodes to store a copy of just one segment at a time. In this particular model, blocks serve as the input for updating the ledger's state. When a block is accepted, the transactions within it are executed, leading to the derivation of a new state based on the previous one. A fixed number of blocks are grouped into a segment, and various nodes store these segments. Users retrieve a segment only when they need to access a specific state. Nodes retain the most recent state to facilitate the verification of new transactions, and they are also assigned a segment by the system to participate in the mining process. Nevertheless, it's important to acknowledge a potential threat from adversaries in this model. These opponents might attempt to permanently disrupt a particular blockchain segment by storing all copies of it and then vanishing from the network once they have succeeded in their efforts.

Zamani et al. [57] put forth RapidChain, a resilient public blockchain protocol designed to withstand Byzantine faults. RapidChain employs a strategy of dividing the node set into numerous smaller groups,

known as committees. These committees function concurrently on separate sets of transaction blocks, each maintaining its distinct ledger. This approach, which involves distributing operations and data across multiple node groups, is commonly referred to as sharding. RapidChain introduces an innovative method for partitioning the blockchain, ensuring that each node is only required to store a fraction of the entire blockchain, specifically, a 1/k portion of it. RapidChain establishes k committees, each comprising n/m nodes, where m = c log n, with c representing a constant determined solely by the security parameter, and n indicating the number of participants. To enable the verification of transactions that span across different shards, RapidChain's committees employ an efficient routing mechanism inspired by Kademlia. This mechanism incurs minimal latency and storage requirements, logarithmic to the number of committees involved.

Jia et al. [58] introduced the concept of ElasticChain, which tackles the challenge of maintaining data safety within a blockchain while reducing the memory load on full nodes. This approach involves partitioning the entire blockchain into segments distributed across nodes, using a duplicate ratio regulation algorithm to ensure data security. This design significantly decreases the memory requirements for full nodes. This algorithm stipulates that full nodes with limited storage capacity are not required to store the complete blockchain; instead, they only retain specific segments of it. The rationale behind this strategy is based on the observation that as blocks become more fundamental, their vulnerability to tampering decreases, thus enhancing the overall security of the blockchain. The degree of duplication for each block is intricately linked to its position within the blockchain. Blocks closer to the foundational components of the chain have fewer duplicates stored, bolstering their resistance to tampering and overall security. In contrast, newer blocks, considered more susceptible, are replicated more extensively to ensure their integrity.

Luu et al. [59] proposed a distributed agreement protocol designed for permission-less blockchains, known as ELASTICO. ELASTICO accomplishes the uniform partitioning or parallelization of the mining network securely into smaller committees. Each committee is responsible for processing a separate set of transactions, often referred to as "shards." In particular, the number of committees expands in close proportion to the overall computational power of the network. Each committee consists of a relatively modest number of members, enabling them to execute a conventional Byzantine consensus protocol concurrently to determine their collectively agreed-upon set of transactions.

Raman et al. [60] introduced a method for dividing participants in blockchain networks into distinct groups referred to as "zones." In this approach, a single instance of each data block is distributed across each of these zone sets, each of which consists of a fixed number of peers, denoted as "zones" and having a size m (set of messages of arbitrary lengths). The process begins with the generation of a private key within each zone, which is then used to encrypt the data block. These private keys are stored by peers within the same zone, employing Shamir's secret key-sharing scheme. Subsequently, the encrypted data block is disseminated among the peers in the zone through a distributed storage mechanism. The allocation of peers to zones dynamically takes place using a technique known as the "m-way Handshake Problem." This method ensures that, over time, every compromised or corrupted peer becomes associated with an uncorrupted peer within the system.

Yin et al. [61] introduced EBSF, a block storage framework that optimizes storage through block allocation plans based on node characteristics. Committees of blockchain nodes collaborate to manage data. Heuristic algorithms address block allocation challenges, including new block arrivals, old block pruning, and node changes, with full and partial allocation strategies.

Table 8 Summary of sharding approach for blockchain storage optimization

Authors	Method	Weakness
Mizrahi et al. [53]	State sharding in blockchain, dividing it into managed shards, minimizing communication with space-aware grouping	Increases the communication overhead
Cai et al. [54]	MaOEA-DRP	It addresses malicious node aggregation but overlooks privacy effects from anonymous address clustering.
Jia et al. [55]	ELM categorizes blocks as hot or non-hot based on popularity and storage.	It requires nodes to have a certain amount of computing power to train and use the ELM classifier

Authors	Method	Weakness
Xu et al. [56]	Introduced segmented blockchain, storing one segment per node for efficient state updates and mining participation	Adversaries aim to disrupt a blockchain segment by storing and disappearing.
Zamani et al. [57]	RapidChain divides nodes into committees, each working on separate transaction blocks and maintaining its unique ledger concurrently.	The initial setup cost is significant.
Jia et al. [58]	ElasticChain, partitions the entire blockchain into segments distributed across nodes using duplicate ratio regulation algorithm	Adds a new chain for storing dependable data, leading to higher communication overhead
Luu et al. [59]	ELASTICO, is a protocol for secure, uniform partitioning of permission-less blockchains into smaller committees, each processing separate transactions (shards).	Complicated architecture
Raman et al. [60]	Creation of blockchain zones, and distributed data with private keys, using Shamir's scheme	Peers' inactivity or data loss hampers recovery.
Yin et al. [61]	EBSF optimizes block storage via node-based allocation, committees, and heuristic algorithms for dynamic blockchain scenarios.	EBSF is a complex system

3.2Off-ledger storage optimization

This approach aims to improve the scalability and efficiency of the blockchain by moving certain data off-ledger, onto external storage or processing solutions. Off-ledger storage optimization offers a way to address these challenges by selectively moving some data away from the main blockchain. Off-ledger storage optimization involves utilizing external storage solutions, such as distributed storage networks and cloud services, to store certain data and files off the main blockchain. *Figure 6* presents the distribution of research articles concentrating on off-ledger storage optimization concerning the total number of research articles reviewed for each respective year. Furthermore, it offers insights into the evolving trends in research and literature within this domain over time.

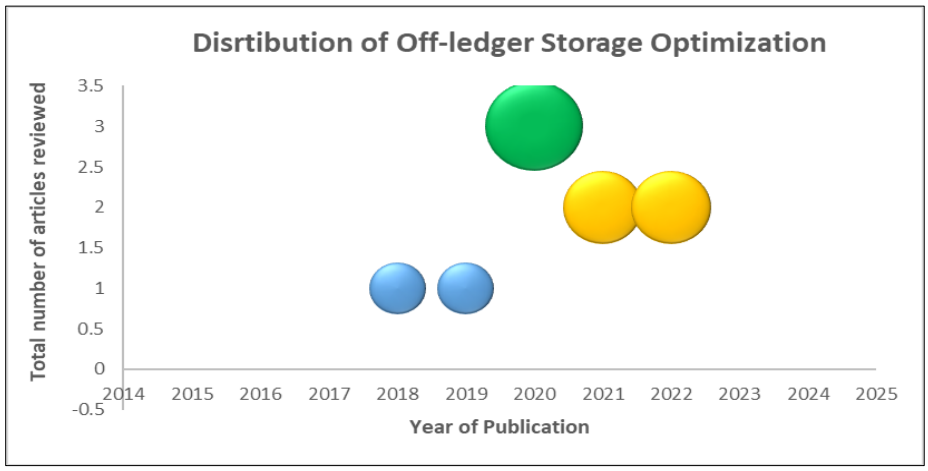

Figure 6 Distribution of on-ledger storage optimization

3.2.1Distributed storage

Distributed storage networks are decentralized systems designed for the storage of data across a network of nodes. They commonly employ technologies such as peer-to-peer networking and encryption. Examples of distributed storage networks include IPFS and distributed hash table (DHT). IPFS is a decentralized storage protocol that operates on a peer-to-peer basis. It breaks down files into uniform-sized blocks and distributes them across various nodes within the network. In a DHT, each key-value pair is linked to a unique identifier, often a hash of the key itself. Nodes within the network take on the responsibility of storing and managing these key-value pairs based on their respective hash values. *Table 9* provides a summary of the optimization of storage through the utilization of distributed storage.

Zhou et al. [62] proposed an innovative approach that IPFS with block compression to minimize the storage demands of the blockchain. This novel scheme operates by preserving the block header within the blockchain while relocating the block body to IPFS. The block header includes crucial information about

the block, such as the previous block's hash, the Merkle root of the transactions, and the difficulty target. Meanwhile, the block body encompasses transaction particulars like sender, receiver, and transaction amount.

Hassanzadeh-nazarabadi et al. [63] introduced LightChain, an innovative blockchain framework that leverages a DHT to enhance scalability. In LightChain, every block and transaction are duplicated within the DHT of peer nodes and is fetched when needed. Consequently, peers in LightChain are not obligated to fetch or maintain the entire ledger. Instead, each peer in LightChain is tasked with preserving a small portion of blocks and transactions, which are randomly assigned. They access other replicated transactions and blocks from peers in the system on an as-needed basis, facilitated by the efficient Skip Graph retrievability mechanism.

Yu et al. [64] proposed a concept known as the virtual block group (VBG) model, which aims to tackle the challenge of node storage scalability. In the context of the VBG model, each node is only required to store a portion of block data and then records the VBG storage index within a DHT, treating the block data as a resource. This approach leads to an enhancement in the efficiency of querying block data. The hash table is subdivided into numerous smaller segments, which are distributed across all nodes within the P2P network. When there is a request for VBG, block, or transaction data through a node, the specific content requested is routed to the node that contains the relevant <key, value> pairs associated with the VBG being requested.

Chen et al. [65] categorized the foundational data of the consortium blockchain into two distinct types: continuous data and state data. They put forth a novel storage architecture called HyperBSA to efficiently handle and manage these two data types. For continuous data, which exhibits a continuous key increase pattern (examples block data, transaction receipts, and transaction logs), they devised an index-based storage engine named Filelog. This engine stores continuous data as files and utilizes a double-layer index approach for rapid data access. Furthermore, it employs a sparse index and dynamically adjusts the step size to cater to the varied random-read requirements of different storage scenarios. As for state data, which lacks the characteristic of continuous key patterns (like account balances), they designed multi-level caching mechanisms to enhance both reading and writing performance.

Zheng et al. [66] introduced a blockchain data storage approach that relies on the IPFS. In this model, miners deposit transaction data into an IPFS network and then incorporate the resulting IPFS hash of the transaction into a block. In this setup, each miner places valid transactions into a memory pool and stores them in IPFS while retaining the corresponding transaction IPFS hash. During the computation of the next block, each miner compiles the IPFS hashes of the verified transactions into a new block and computes both the Merkle root and the block hash. If a miner successfully calculates a block hash that meets the specified difficulty level, the block is then broadcasted to all nodes within the blockchain network.

Table 9 Summary of distributed storage approach for blockchain storage optimization

Authors	Method	% of Storage Reduction	Weakness
Zhou et al. [62]	IPFS and block compression to reduce blockchain storage. The block header remains, while the block body moves to IPFS.	96.9%	It requires all full nodes to run the IPFS network
Hassanzadeh-nazarabadi et al. [63]	LightChain uses a DHT in which peers store and share data efficiently.	Storage per node is reduced by 66 times.	Low transaction throughput
Yu et al. [64]	VBG which shares block data efficiently in a DHT.	--	Higher expenses for retrieving block data from remote sources.
Chen et al. [65]	HyperBSA storage architecture managing continuous and state data with Filelog and caching mechanisms.	--	Enhances ledger read-write performance but neglects blockchain scalability.
Zheng et al. [66]	Utilized IPFS for blockchain data storage, depositing transaction data and hashes into blocks for consensus	91.83%	Increased delay resulting from queries made to the IPFS network.

3.2.2Cloud storage

In this technique, data is stored outside of the blockchain ledger, in a cloud storage provider. This can be done to reduce the storage requirements of the blockchain, improve performance, or increase scalability. *Table 10* shows a summary of storage optimization using cloud storage.

Xu et al. [67] introduced the futile transactions filter (FTF), which is a transaction filtering and offloading system aimed at reducing the storage and communication burden of the blockchain. This scheme identifies and transfers futile transactions to the cloud, which are transactions that hold no relevance for the edge devices. In essence, futile transactions are those whose outputs are entirely referenced by subsequent transactions and do not contribute to the validation of newly generated transactions. FTF systematically reviews all the transactions, categorizing them as either useful (non-futile) or futile. Once FTF completes the futile transaction filtering process, the transaction offloading module pinpoints the blocks containing valuable transactions and updates them to the cache layer. Conversely, the futile blocks, which exclusively contain futile transactions, are dispatched to stakeholder clouds for backup. Subsequently, these blocks are removed from the edge devices, thereby optimizing storage efficiency.

Koshy et al. [68] proposed a concept called the sliding window blockchain (SWBC), which employs a moving window that traverses through the blockchain with each new block addition. This window initially encompasses a single block and expands gradually, reaching a size of n blocks as defined by the window's dimensions. The blocks contained within this sliding window are utilized during the creation of a new block. Within the proposed SWBC framework, the block hash is computed by hashing the blocks residing within the window. The size of the sliding window determines how many recent past blocks are involved in this hash update process. The most recent n blocks are stored in the memory of IoT devices, while the entire blockchain is maintained within a private cloud. As the window slides, older blocks exit the window and are subsequently removed from the memory of IoT devices. This approach helps manage and optimize storage effectively.

Zhang et al. [69] presented a strategy that categorizes blockchain transaction databases into two segments: cold zones and hot zones, employing the least recently used (LRU) algorithm for recognizing transaction expirations. This approach aims to optimize storage by relocating UTXOs from the in-memory transaction databases. In this context, every blockchain node maintains an in-memory UTXO set, primarily for generating and verifying new transactions. However, when private keys are lost, certain UTXOs become permanently unused and continue residing in the in-memory UTXO set. This paper introduces a method to enhance UTXO storage efficiency, guided by the LRU algorithm. The scheme establishes an expiration policy and identifies regularly expiring UTXOs. When an expired UTXO is identified, it is transferred out of the active node's memory, thereby increasing the available memory space.

Liao et al. [70] introduced the graph partition-based storage strategy for DAG-Blockchain (GpDB). GpDB introduces a graph partitioning algorithm that takes into account the freshness of transactions. This algorithm divides the topology of the DAG blockchain into two segments within edge servers: one segment is retained, and the other is discarded. The cloud server is tasked with preserving the entire ledger data, while the edge server's responsibility is limited to storing a portion of the data and maintaining a consistent transaction storage cost. This approach effectively optimizes the storage expenses associated with edge servers.

Table 10 Summary of cloud storage approach for blockchain storage optimization

Authors	Method	Weakness
Xu et al. [67]	FTF to reduce blockchain storage and communication load by offloading irrelevant transactions to the cloud.	FTF implementation is more complex and also requires analysis of uncle blocks for Ethereum.
Koshy et al. [68]	SWBC, uses a moving window to compute block hashes and optimize storage.	The computation time grows linearly with the increase in window size
Zhang et al. [69]	Relocating UTXOs using the LRU algorithm	Data query calls to the cold zone are inefficient
Liao et al. [70]	GpDB, a storage strategy for DAG blockchains using graph partitioning based on transaction freshness	DAG-blockchain storage nodes possess varying storage capacities, resulting in an imbalance

4.Discussion

Methods employed for on-chain storage, including compression, summarization, block pruning, and partial node storage, introduce challenges related to trackability, accuracy, integrity, and data consistency due to potential data loss. The utilization of sharding

exacerbates complexity and raises concerns about data consistency. *Table 11* illustrates the challenges encountered in optimizing on-chain storage optimization.

Table 11 Challenges encountered in optimizing on-chain storage optimization

On-chain storage optimization method	Description	Challenges
Compression, Summarization, Partial Node Storage, Block Pruning	• Integrity problems because of data loss	• The removal of original data and block compression make it harder to verify blockchain integrity.
		• Verifying data on blockchain can be difficult due to the loss of granularity and detail.
		• Storing only a portion of blockchain data increases risk of data loss and Security.
		• Pruning blocks increasing risk of data loss.
Sharding	• Increases complexity	• Sharding increases complexity, security risks, data consistency issues, and network fragmentation.

Therefore, off-chain storage optimization can be effectively used to optimize the storage capacity of peers improving the scalability and reliability of the blockchain network. Distributed storage solutions like DHT and IPFS are complex, expensive, and encounter compatibility issues. Therefore, cloud storage can be efficiently employed to enhance data storage optimization. To address the storage limitations, the optimal blocks selected using multi-optimization algorithm can be stored in the cloud, providing peers with ample storage capacity. The techniques such as LRU and SWBC aim to identify and store crucial blocks in the cloud for future use, but this approach can lead to higher cloud expenses. To address this, multi-objective optimization algorithms become valuable as they can efficiently choose blocks considering factors like query probability, cloud storage cost, and local space usage. By employing these algorithms, one can select the optimal blocks with the least cloud storage cost and the highest local space occupancy.

4.1.1Blockchain storage optimization leveraging cloud storage and multi-objective optimization algorithms

Xu et al. [71] proposed a method to enhance the storage capacity of individual peers in a blockchain network by storing less frequently accessed old blocks in the cloud. They proposed a system in which network peers are linked to cloud servers, and the chosen blocks are transferred to these servers. The authors formulated the problem of selecting blocks using a multi-objective optimization problem with objective functions related to the likelihood of being queried, storage costs, and local space usage. To tackle this problem, they proposed a nondominated sorting genetic algorithm with clustering (NSGA-C). The authors tested their proposed method using a blockchain network for an IoT application. The network had three peers and 200 blocks, each of which was 1MB in size. However, the algorithm had a longer runtime than the benchmark algorithms.

Nartey et al. [72] built upon their previous research [14] by introducing an enhanced approach. They proposed an advanced time variant multi-objective particle swarm optimization (AT-MOPSO) algorithm to address the block selection problem considering factors like query probability, cloud storage cost, and local space occupancy. Their work focused on a blockchain-IIoT framework that utilized fog nodes running side chains with containerization for block selection. Despite improvements in the runtime and energy efficiency, the AT-MOPSO algorithm did not outperform NSGA-C in terms of the local space occupancy objective. As a result, there may be limitations in reducing storage overhead for peers. The proposed approach of the authors was examined by conducting experiments on an IoT application using a blockchain network. This network consisted of three peers and included 200 blocks, with each block being 1MB in size.

Akrasi-mensah et al. [73] proposed an adaptive optimization scheme for blockchain-IIoT systems that leverages cloud storage to reduce storage demand on local peers. To find the optimal set of blocks that should be kept in cloud storage, authors proposed using a deep reinforcement learning (DRL) agent that learns from interacting with the blockchain and evaluating the parameters such as query probability, cloud storage cost, and local space occupancy of the blocks to discover which blocks should be selected. This research has the challenge of ensuring the integrity and security of the blockchain when blocks are transferred to and from the cloud. The authors conducted experiments on an IoT application by implementing their suggested technique on a

blockchain network. The network consisted of three peers, and it comprised a total of 200 blocks, with each block having a size from 1KB to 1MB.

Zhou et al. [74] proposed a new algorithm for optimizing the storage of blockchain data in the cloud. The proposed algorithm is based on a combination of the non-dominated sorting genetic algorithm III (NSGA-III) and deep q-networks (DQN). In order to assess their proposed method, the authors utilized a blockchain network tailored for an Internet of Things (IoT) application. The network was composed of three peers and encompassed a collection of 200 blocks, with each block having a size of 1MB. This method has outperformed other approaches in terms of block selection.

4.1.2Future scope

The blockchain storage optimization utilizing cloud storage and a multi-objective optimization algorithm, as discussed earlier, has privacy concerns when optimal blocks are stored in the cloud. Therefore, any selected block stored in the cloud must be encrypted. Additionally, the process of selecting optimal blocks via the multi-objective optimization algorithm may result in choosing blocks that contain sensitive data. To address this, blocks should be selected to ensure they do not contain any sensitive information. Furthermore, incorporating a trust level for cloud servers can ensure that the selected data is stored on trusted cloud servers. A complete list of abbreviations is listed in *Appendix I.*

5.Conclusion

Blockchain technology has seen rapid growth in recent years, indicating its increasing use across diverse applications. However, the adoption of blockchain systems presents both advantages and challenges, especially as the number of network nodes expands. A significant challenge lies in the growing storage requirements as the blockchain network scales up. This paper investigates the impact of storage on blockchain scalability and categorizes various storage optimization solutions into two groups: on-ledger and off-ledger approaches. On-ledger methods encompass compression, summarization, block pruning, partial node storage, and sharding, while off-chain approaches involve distributed storage and cloud storage. We then provide a summary of existing solutions and ideas focusing on storage optimization within these on-ledger and off-ledger approaches. Optimizing storage, including techniques like compression, summarization, block pruning, and partial node storage, presents a challenge due to potential data loss. Sharding increases communication overhead while implementing distributed storage solutions like IPFS and DHT can be complex. Utilizing cloud storage is viable only when selecting specific blocks for storage on the cloud, considering factors like cloud storage cost, query likelihood, and local space usage.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

None.

Author's contribution statement

Shelke R. Kavita: Conceptualization, investigation, writing–review and editing. **Dr. Subhash K. Shinde:** Conceptualization, investigation, and supervision.

References

[1] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2008.

[2] Yuan Y, Wang FY. Blockchain and cryptocurrencies: model, techniques, and applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2018; 48(9):1421-8.

[3] Guo H, Yu X. A survey on blockchain technology and its security. Blockchain: Research and Applications. 2022; 3(2):100067.

[4] Abou JJ, Saade RG. Blockchain applications–usage in different domains. IEEE Access. 2019; 7:45360-81.

[5] Khan SN, Loukil F, Ghedira-guegan C, Benkhelifa E, Bani-hani A. Blockchain smart contracts: applications, challenges, and future trends. Peer-to-peer Networking and Applications. 2021; 14:2901-25.

[6] Manzoor R, Sahay BS, Singh SK. Blockchain technology in supply chain management: an organizational theoretic overview and research agenda. Annals of Operations Research. 2022:1-48.

[7] Rejeb A, Treiblmaier H, Rejeb K, Zailani S. Blockchain research in healthcare: a bibliometric review and current research trends. Journal of Data, Information and Management. 2021; 3:109-24.

[8] Jafar U, Aziz MJ, Shukur Z. Blockchain for electronic voting system-review and open research challenges. Sensors. 2021; 21(17):1-22.

[9] Weerawarna R, Miah SJ, Shao X. Emerging advances of blockchain technology in finance: a content analysis. Personal and Ubiquitous Computing. 2023; 27(4):1495-508.

[10] Saari A, Vimpari J, Junnila S. Blockchain in real estate: recent developments and empirical applications. Land Use Policy. 2022; 121:106334.

[11] Ali MS, Vecchio M, Pincheira M, Dolui K, Antonelli F, Rehmani MH. Applications of blockchains in the internet of things: a comprehensive survey. IEEE

Communications Surveys & Tutorials. 2018; 21(2):1676-717.
[12] Choo KK, Ozcan S, Dehghantanha A, Parizi RM. Blockchain ecosystem—technological and management opportunities and challenges. IEEE Transactions on Engineering Management. 2020; 67(4):982-7.
[13] Lu Y. The blockchain: state-of-the-art and research challenges. Journal of Industrial Information Integration. 2019; 15:80-90.
[14] Uddin MA, Stranieri A, Gondal I, Balasubramanian V. A survey on the adoption of blockchain in IoT: challenges and solutions. Blockchain: Research and Applications. 2021; 2(2):100006.
[15] Habib G, Sharma S, Ibrahim S, Ahmad I, Qureshi S, Ishfaq M. Blockchain technology: benefits, challenges, applications, and integration of blockchain technology with cloud computing. Future Internet. 2022; 14(11):1-22.
[16] Mazlan AA, Daud SM, Sam SM, Abas H, Rasid SZ, Yusof MF. Scalability challenges in healthcare blockchain system-a systematic review. IEEE Access. 2020; 8:23663-73.
[17] Zhou Q, Huang H, Zheng Z, Bian J. Solutions to scalability of blockchain: a survey. IEEE Access. 2020; 8:16440-55.
[18] Xie J, Yu FR, Huang T, Xie R, Liu J, Liu Y. A survey on the scalability of blockchain systems. IEEE Network. 2019; 33(5):166-73.
[19] Khan D, Jung LT, Hashmani MA. Systematic literature review of challenges in blockchain scalability. Applied Sciences. 2021; 11(20):1-27.
[20] Kim S, Kwon Y, Cho S. A survey of scalability solutions on blockchain. In international conference on information and communication technology convergence 2018 (pp. 1204-7). IEEE.
[21] https://www.statista.com/statistics/730806/daily-number-of-bitcoin-transactions/. Accessed 15 May 2024.
[22] https://learn.bybit.com/blockchain/bitcoin-mempool-what-happens-to-the-unconfirmed-transactions/. Accessed 15 May 2024.
[23] https://www.statista.com/statistics/730818/average-number-of-ethereum-transactions/. Accessed 15 May 2024.
[24] Akrasi-mensah NK, Tchao ET, Sikora A, Agbemenu AS, Nunoo-mensah H, Ahmed AR, et al. An overview of technologies for improving storage efficiency in blockchain-based IIoT applications. Electronics. 2022; 11(16):1-25.
[25] Kim T, Lee S, Kwon Y, Noh J, Kim S, Cho S. SELCOM: selective compression scheme for lightweight nodes in blockchain system. IEEE Access. 2020; 8:225613-26.
[26] Yu B, Li X, Zhao H. PoW-BC: a PoW consensus protocol based on block compression. KSII Transactions on Internet & Information Systems. 2021; 15(4).
[27] Chen X, Lin S, Yu N. Bitcoin blockchain compression algorithm for blank node synchronization. In 11th international conference on wireless communications and signal processing (WCSP) 2019 (pp. 1-6). IEEE.
[28] Qi S, Lu Y, Zheng Y, Li Y, Chen X. Cpds: enabling compressed and private data sharing for industrial internet of things over blockchain. IEEE Transactions on Industrial Informatics. 2020; 17(4):2376-87.
[29] Guo Z, Gao Z, Liu Q, Chakraborty C, Hua Q, Yu K, et al. RNS-based adaptive compression scheme for the block data in the blockchain for IIoT. IEEE Transactions on Industrial Informatics. 2022; 18(12):9239-49.
[30] Marsalek A, Zefferer T, Fasllija E, Ziegler D. Tackling data inefficiency: compressing the bitcoin blockchain. In 18th international conference on trust, security and privacy in computing and communications/13th international conference on big data science and engineering (trustcom/bigdatase) 2019 (pp. 626-33). IEEE.
[31] Xu L, Chen L, Gao Z, Xu S, Shi W. EPBC: efficient public blockchain client for lightweight users. In proceedings of the 1st workshop on scalable and resilient infrastructures for distributed ledgers 2017 (pp. 1-6). ACM.
[32] Dorri A, Kanhere SS, Jurdak R. MOF-BC: a memory optimized and flexible blockchain for large scale networks. Future Generation Computer Systems. 2019; 92:357-73.
[33] Nadiya U, Mutijarsa K, Rizqi CY. Block summarization and compression in bitcoin blockchain. In international symposium on electronics and smart devices 2018 (pp. 1-4). IEEE.
[34] Palai A, Vora M, Shah A. Empowering light nodes in blockchains with block summarization. In 9th IFIP international conference on new technologies, mobility and security 2018 (pp. 1-5). IEEE.
[35] Shi D, Wang X, Xu M, Kou L, Cheng H. Ress: a reliable and efficient storage scheme for bitcoin blockchain based on raptor code. Chinese Journal of Electronics. 2023; 32(3):577-86.
[36] Qi X, Zhang Z, Jin C, Zhou A. A reliable storage partition for permissioned blockchain. IEEE Transactions on Knowledge and Data Engineering. 2020; 33(1):14-27.
[37] Liu Y, Wang K, Lin Y, Xu W. $\mathsf {LightChain} $: a lightweight blockchain system for industrial internet of things. IEEE Transactions on Industrial Informatics. 2019; 15(6):3571-81.
[38] Li C, Zhang J, Yang X, Youlong L. Lightweight blockchain consensus mechanism and storage optimization for resource-constrained IoT devices. Information Processing & Management. 2021; 58(4):102602.
[39] Chen J, Lv Z, Song H. Design of personnel big data management system based on blockchain. Future Generation Computer Systems. 2019; 101:1122-9.
[40] Mei H, Gao Z, Guo Z, Zhao M, Yang J. Storage mechanism optimization in blockchain system based on residual number system. IEEE Access. 2019; 7:114539-46.

[41] Zhao Y, Niu B, Li P, Fan X. A novel enhanced lightweight node for blockchain. In blockchain and trustworthy systems: first international conference, blocksys 2019, Guangzhou, China, 2019, Proceedings 1 2020 (pp. 137-49). Springer Singapore.
[42] Xu Z, Han S, Chen L. CUB, a consensus unit-based storage scheme for blockchain system. In 34th international conference on data engineering 2018 (pp. 173-84). IEEE.
[43] Xu Y. Section-blockchain: a storage reduced blockchain protocol, the foundation of an autotrophic decentralized storage architecture. In 23rd international conference on engineering of complex computer systems 2018 (pp. 115-25). IEEE.
[44] Li D, Dai J, Jiang R, Wang X, Xu Y. GAPG: a heuristic greedy algorithm for grouping storage scheme in blockchain. In IEEE/CIC international conference on communications in China (ICCC Workshops) 2020 (pp. 91-5). IEEE.
[45] Liu T, Wu J, Li J, Li J. Secure and balanced scheme for non-local data storage in blockchain network. In 21st international conference on high performance computing and communications; 17th international conference on smart city; 5th international conference on data science and systems (HPCC/SmartCity/DSS) 2019 (pp. 2424-7). IEEE.
[46] Dai X, Xiao J, Yang W, Wang C, Jin H. Jidar: a jigsaw-like data reduction approach without trust assumptions for bitcoin system. In 39th international conference on distributed computing systems 2019 (pp. 1317-26). IEEE.
[47] Wang X, Wang C, Zhou K, Cheng H. Ess: an efficient storage scheme for improving the scalability of bitcoin network. IEEE Transactions on Network and Service Management. 2021; 19(2):1191-202.
[48] Matzutt R, Kalde B, Pennekamp J, Drichel A, Henze M, Wehrle K. Coinprune: shrinking bitcoin's blockchain retrospectively. IEEE Transactions on Network and Service Management. 2021; 18(3):3064-78.
[49] Heo JW, Ramachandran GS, Dorri A, Jurdak R. Blockchain storage optimisation with multi-level distributed caching. IEEE Transactions on Network and Service Management. 2022; 19(4):3724-36.
[50] Pyoung CK, Baek SJ. Blockchain of finite-lifetime blocks with applications to edge-based IoT. IEEE Internet of Things Journal. 2019; 7(3):2102-16.
[51] Gao J, Li B, Li Z. Blockchain storage analysis and optimization of bitcoin miner node. In communications, signal processing, and systems: proceedings of the 2018 CSPS volume III: systems 7th 2020 (pp. 922-32). Springer Singapore.
[52] Chen H, Wang Y. MiniChain: a lightweight protocol to combat the UTXO growth in public blockchain. Journal of Parallel and Distributed Computing. 2020; 143:67-76.
[53] Mizrahi A, Rottenstreich O. Blockchain state sharding with space-aware representations. IEEE Transactions on Network and Service Management. 2020; 18(2):1571-83.
[54] Cai X, Geng S, Zhang J, Wu D, Cui Z, Zhang W, et al. A sharding scheme-based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial internet of things. IEEE Transactions on Industrial Informatics. 2021; 17(11):7650-8.
[55] Jia D, Xin J, Wang Z, Wang G. Optimized data storage method for sharding-based blockchain. IEEE Access. 2021; 9:67890-900.
[56] Xu Y, Huang Y. Segment blockchain: a size reduced storage mechanism for blockchain. IEEE Access. 2020; 8:17434-41.
[57] Zamani M, Movahedi M, Raykova M. Rapidchain: scaling blockchain via full sharding. In proceedings of the SIGSAC conference on computer and communications security 2018 (pp. 931-48). ACM.
[58] Jia D, Xin J, Wang Z, Guo W, Wang G. ElasticChain: support very large blockchain by reducing data redundancy. In web and big data: second international joint conference, APWeb-WAIM 2018, Macau, China, 2018, Proceedings, Part II 2 2018 (pp. 440-54). Springer International Publishing.
[59] Luu L, Narayanan V, Zheng C, Baweja K, Gilbert S, Saxena P. A secure sharding protocol for open blockchains. In proceedings of the SIGSAC conference on computer and communications security 2016 (pp. 17-30). ACM.
[60] Raman RK, Varshney LR. Dynamic distributed storage for blockchains. In international symposium on information theory 2018 (pp. 2619-23). IEEE.
[61] Yin B, Li J, Wei X. EBSF: node characteristics-based block allocation plans for efficient blockchain storage. IEEE Transactions on Network and Service Management. 2022; 19(4):4858-71.
[62] Zhou K, Wang C, Wang X, Chen S, Cheng H. A novel scheme to improve the scalability of bitcoin combining ipfs with block compression. IEEE Transactions on Network and Service Management. 2022; 19(4):3694-705.
[63] Hassanzadeh-nazarabadi Y, Küpçü A, Özkasap Ö. LightChain: scalable DHT-based blockchain. IEEE Transactions on Parallel and Distributed Systems. 2021; 32(10):2582-93.
[64] Yu B, Li X, Zhao H. Virtual block group: a scalable blockchain model with partial node storage and distributed hash table. The Computer Journal. 2020; 63(10):1524-36.
[65] Chen X, Zhang K, Liang X, Qiu W, Zhang Z, Tu D. HyperBSA: a high-performance consortium blockchain storage architecture for massive data. IEEE Access. 2020; 8:178402-13.
[66] Zheng Q, Li Y, Chen P, Dong X. An innovative IPFS-based storage model for blockchain. In IEEE/WIC/ACM international conference on web intelligence 2018 (pp. 704-8). IEEE.
[67] Xu C, Wang K, Li P, Guo S, Luo J, Ye B, Guo M. Making big data open in edges: a resource-efficient blockchain-based approach. IEEE Transactions on Parallel and Distributed Systems. 2018; 30(4):870-82.

[68] Koshy P, Babu S, Manoj BS. Sliding window blockchain architecture for internet of things. IEEE Internet of Things Journal. 2020; 7(4):3338-48.
[69] Zhang J, Zhong S, Wang J, Yu X, Alfarraj O. A storage optimization scheme for blockchain transaction databases. Computer Systems Science & Engineering. 2021; 36(3):521-35.
[70] Liao Z, Cheng S, Zhang J, Wu W, Wang J, Sharma PK. GpDB: a graph-partition based storage strategy for DAG-blockchain in edge-cloud IIoT. IEEE Transactions on Industrial Informatics. 2022.
[71] Xu M, Feng G, Ren Y, Zhang X. On cloud storage optimization of blockchain with a clustering-based genetic algorithm. IEEE Internet of Things Journal. 2020; 7(9):8547-58.
[72] Nartey C, Tchao ET, Gadze JD, Yeboah-akowuah B, Nunoo-mensah H, Welte D, et al. Blockchain-IoT peer device storage optimization using an advanced time-variant multi-objective particle swarm optimization algorithm. EURASIP Journal on Wireless Communications and Networking. 2022; 2022(1):5.
[73] Akrasi-mensah NK, Agbemenu AS, Nunoo-mensah H, Tchao ET, Ahmed AR, Keelson E, et al. Adaptive storage optimization scheme for blockchain-IIoT applications using deep reinforcement learning. IEEE Access. 2022; 11:1372-85.
[74] Zhou Y, Ren Y, Xu M, Feng G. An improved NSGA-III algorithm based on deep Q-networks for cloud storage optimization of blockchain. IEEE Transactions on Parallel and Distributed Systems. 2023; 34(5):1406-19.

Shelke R. Kavita is a Research Scholar and Assistant Professor in Computer Engineering at Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai. She has completed her M.E. Computer Engineering in 2013. She has more than 15 years' experience in the field of academics. She has published about 18 papers in International Journals and Conferences. Her research area includes Blockchain, Artificial Intelligence, Computational Algorithms.
Email: kavita.shelke@fcrit.ac.in

Dr. Subhash K. Shinde an accomplished Computer Engineer, is the Principal and Professor in Computer Engineering at Lokmanya Tilak College of Engineering, Navi Mumbai. He has completed his M.E. in Information Technology (1999) and Ph.D. in Computer Science and Engineering (2012). He has more than 24 years' experience in the field of academics, administration and research. He has published about 55 papers in International Journals and Conferences and has a copyright to his credit. He has also authored five books through reputed publishers. Under his supervision, 6 research scholars (PhD) and 30 PG (ME) Students have successfully completed their degrees from the University of Mumbai. He is working as a Chairman Board of Studies in Computer Engineering under the Faculty of Technology, University of Mumbai, and a Member of the Academic Council, BUTR, RRC, of the University of Mumbai from Nov 2023.
Email: skshinde@rediffmail.com

Appendix I

S. No.	Abbreviation	Description
1	AF	Access Frequency
2	AT-MOPSO	Advanced Time Variant Multi-Objective Particle Swarm Optimization
3	BFT	Byzantine Fault Tolerance
4	BMT	Block Merkle Tree
5	CNs	Computing Nodes
6	Cpds	Compressed and Private Data Sharing
7	CRT	Chinese Remainder Theorem
8	CRT-II	Chinese Remainder Theorem-II
9	CU	Consensus Unit
10	dApps	Decentralized Applications
11	DHT	Distributed Hash Table
12	DNs	Demanding Nodes
13	DRL	Deep Reinforcement Learning
14	DQN	Deep Q-Networks
15	ELM	Extreme Learning Machine
16	EPBC	Efficient Public Blockchain Client
17	ESPV	Enhanced Simplified Payment Verification
18	ESS	Effective Storage Strategy
19	FMA	Fast-Matching Algorithm
20	FTF	Futile Transactions Filter
21	GA	Genetic Algorithm
22	IoT	Internet of Things
23	IIoT	Industrial Internet of Things
24	IP	Internet Protocol
25	IPFS	Inter Planetary File System
26	Jidar	Jigsaw-like Data Reduction approach
27	LRU	Least Recently Used
28	MaOEA-DRP	Many-Objective Optimization Algorithm- Dynamic Reward and Penalty Mechanism
29	MMR	Merkle Mountain Range
30	MLDC	Multi-Level Distributed Caching
31	MOFBC	Memory Optimized and Flexible Blockchain
32	NSGA-C	Nondominated Sorting Genetic Algorithm with Clustering
33	NSGA-III	Non-dominated Sorting Genetic Algorithm
34	PNs	Providing Nodes
35	PRISMA	Preferred Reporting Items for Systematic Review and Meta-Analysis
36	R-ABC	Residue Number System Based Adaptive Compression
37	RNS	Residual Number System
38	RS	Reed-Solomon
39	RSA	Rivest–Shamir–Adleman
40	SC	Storage Classification
41	SELCOM	Selective Compression
42	STXO	Spent Transaction Outputs
43	SWBC	Sliding Window Blockchain
44	TSA	Tabu Search Algorithm
45	TXO	Transaction Outputs
46	UBs	Unrelated Blocks
47	UBOF	Unrelated Block Offloading Filter
48	UTXO	Unspent Transaction Output
49	VBG	Virtual Block Group