Cyber Security and Applications 3 (2025) 100065

Contents lists available at ScienceDirect

Cyber Security and Applications

journal homepage: http://www.keaipublishing.com/en/journals/cyber-security-and-applications/

A survey on scalable consensus algorithms for blockchain technology

Ankit Kumar Jain [a], Nishant Gupta [a], Brij B. Gupta [b,c,d,*]

[a] National Institute of Technology Kurukshetra, India
[b] Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan, China
[c] Symbiosis Centre for Information Technology (SCIT), Symbiosis International University, Pune, India
[d] Center for Interdisciplinary Research, University of Petroleum and Energy Studies (UPES), Dehradun, India

ARTICLE INFO

Keywords:
Consensus algorithm
Blockchain technology
Byzantine faults
Throughput
Latency

ABSTRACT

The process of reaching an agreement on a value within a distributed network, known as a consensus problem, is a defining feature of blockchain. This consensus problem can be seen in various applications like load balancing, transaction validation in blockchain, and distributed computing. In recent years, many researchers have provided solutions to this problem. Hence we have presented a survey in which we delved into blockchain consensus algorithms and conducted a comparative analysis of all the consensus algorithms to provide information about each protocol's advantages and drawbacks. This survey starts with the standard proof-of-work consensus protocol applied in bitcoin cryptocurrency and its limitations on the ground of the following parameters: throughput (transactions per second), latency, forks, fault tolerance, double spending attacks, and power consumption. The rest of the consensus algorithms in this paper have been systematically covered to address the limitations of proof-of-work. This paper also covered Raft and PBFT consensus algorithms suitable for permissioned networks. Although the PBFT consensus protocol has a high throughput and a low latency, it has limited node scalability. The PBFT has a low byzantine fault tolerant rate. This paper also covers PoEWAL for blockchain-based IoT applications and WBFT, which prevents corrupt nodes from taking part in consensus. A comparative analysis of the consensus algorithms provides an explicit knowledge of the present research, which also offers guidance for future study.

1. Introduction

A blockchain is like a digital ledger or a database that stores information in a way that makes it almost impossible to tamper with or alter. A consensus algorithm allows people or computers to agree on something together. It is like trying to decide on a pizza topping with your friends. Everyone has different preferences, but you all need to agree on something to order a pizza. In 2008, Nakamoto et al. [1] presented a digital currency that uses the proof-of-work method for transactions on a peer-to-peer network without needing a third person. The digital currency bitcoin was the first implementation of blockchain technology. The success of bitcoin and blockchain offers the following properties: transparency, trustworthiness, dependability, immutability, traceability, decentralization, and tamper-proof have encouraged other industries to adopt blockchain technology. Consensus protocol, cryptographic hash, Distributed ledger, and smart contracts are only a few of the principles used in blockchain. The consensus protocol is the most significant idea among them all. The consensus protocol is the foundation of blockchain technology. The blockchain network's security and integrity are upheld via the consensus protocol [2]. In Section 1.2, the consensus protocol is covered in more detail. The main barrier to blockchain adoption in various industries is scalability issues. The main three scalability issues are limited block size, high computational requirements, and high network latency. Many blockchain networks have limited block sizes which results in low throughput. Blockchain uses consensus protocols. The consensus protocol, like proof-of-work, requires high computation, leading to high electricity consumption. Many blockchain networks have high latency. Latency means the delay in time to commit a transaction in a blockchain network [3]. In Section 4, these scalability concerns are covered in further depth.

1.1. Blockchain technology

Decentralization, or the lack of reliance on a central authority, is one of the properties of the blockchain. Decentralization implies that no single authority dominates the entire bitcoin network; rather, it is managed by a group of servers or computers called "nodes" or "peers". These nodes validate all transactions occurring in the bitcoin system.

Peer review under responsibility of KeAi Communications Co., Ltd.
* Corresponding author.
E-mail address: bbgupta@asia.edu.tw (B.B. Gupta).

https://doi.org/10.1016/j.csa.2024.100065
Available online 8 July 2024

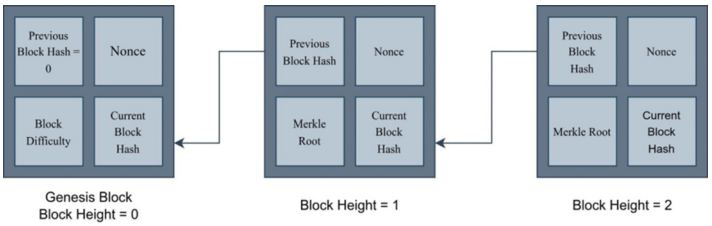

Fig. 1. The figure shows the chain of blocks connected to form a blockchain.

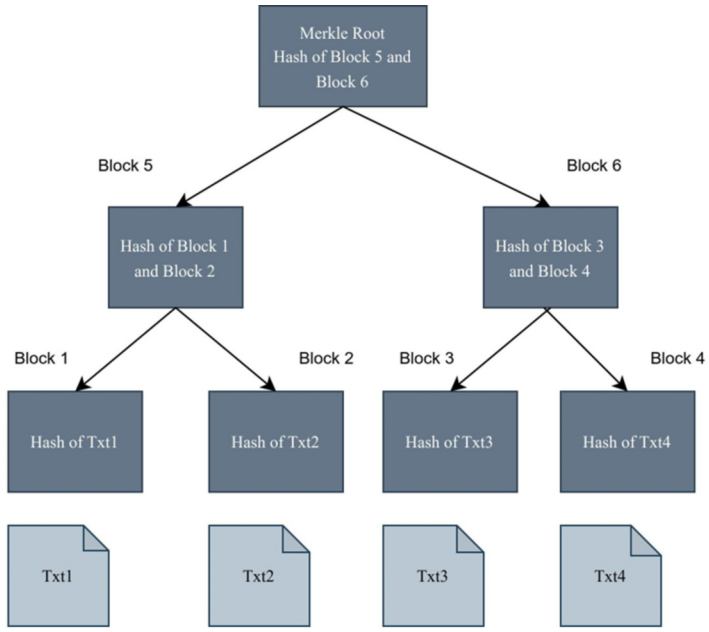

Fig. 2. Merkle Tree. Txt = Transaction.

Blockchain, as a decentralized system, has motivated others to use this new technology who do not want a single authority to control all their transactions [4]. One can define blockchain as a "distributed ledger" replicated across network participants. Network participants maintain the distributed ledger that keeps track of all network transactions. The information or data stored on the ledger is append-only. This means that it cannot be modified once data is appended to the ledger. This is called the "immutability" property, and it reassures network users that their data is secure. A series of linked blocks make up a blockchain. Each block contains a nonce, the previous block's hash, the timestamp, the Merkle Root, the block height, the current block's hash, and a collection of transactions, as seen in Fig. 1. The preceding block's hash is set to zero in the genesis block. Blockchain difficulty is also contained in the genesis block.

Hash: The hash function is used to calculate the hash value. The one-way function is characteristic of the hash function. Here, the one-way function means one can always determine the same output from the given input, however, it is mathematically tough to infer the input from the provided result [5]. The hash function's input is of variable length, but the hash function's result is of fixed-size data. The algorithms to calculate hash value are MD5 and SHA-256. SHA-256 is used in bitcoin to calculate the hash value. Because each block's hash value is distinct, the hash value helps identify a particular block ensuring the integrity of data, linking blocks together, and securing the blockchain network [6].

Block Difficulty: The number of prefix zeros in the hash function's result is set before computing the hash value. Block difficulty is the number of prefix zeros in the hash value. The higher difficulty level stops attackers from tampering with the block's data and increases the blockchain's security [7].

Nonce: Nonce is a random number used to compute the specific number of prefix zeros in the hash value. The best method to find the required hash is the random search. Nonce value helps in a random search. The nonce value is combined with the input before applying it to the hash function. The computation of the required hash requires high CPU power and high electricity consumption [8,9].

Block Height: The number of blocks preceding the specified block is called the block height. Since there is no block before the genesis block, its height is 0. The blockchain's height is equal to the block's highest height [10].

Timestamp: Timestamp records the time the block is created [11,12], Timestamps ensure the order of blocks in the blockchain network as well as prevent the possibility of double-spending attacks. Here, double-spending attacks mean spending the same coin twice. Merkle Root: Utilizing a data structure known as a Merkle tree, the hash values of all transactions recorded in a block are combined to form the Merkle root.

The Merkle tree is akin to a binary data structure. The hash value of each leaf node corresponds to a specific transaction, while each non-leaf node contains a hash of its child nodes. At the core of the Merkle tree lies the Merkle root, serving as a summary of all transactions within the block [13]. This Merkle root, which validates the integrity of block transactions without the need to verify individual transactions, directly reflects changes in the block's transactions [14]. Fig. 2 illustrates the Merkle tree data structure.

1.2. *What is a consensus algorithm?*

A distributed network employs a consensus algorithm to achieve agreement on a value. In the realm of blockchain, consensus protocols play a crucial role in reaching an agreement on the block that comprises a set of transactions and in ensuring that network participants unanimously acknowledge the current state of the blockchain. In this context, reaching consensus on a block involves determining the validity of the transactions contained within it. The use of a consensus algorithm is fundamental to guaranteeing the security and integrity of

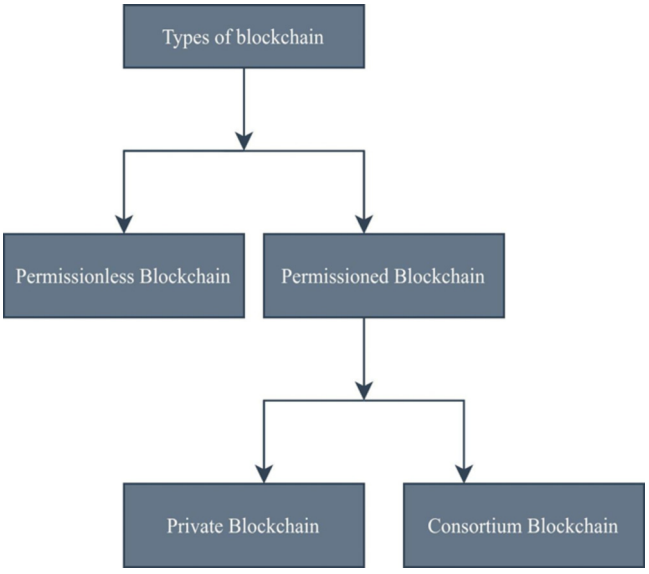

Fig. 3. Types of Blockchain.

the system. Consequently, the core of blockchain technology lies in its consensus protocol [15]. The Bitcoin network uses proof-of-work. The bitcoin network nodes prove that they have done the required work to solve the cryptographic puzzle before proposing the block. The network sets the difficulty of the cryptographic puzzle. The cryptographic puzzle is the required number of leading zeros in the hash value discussed in Section 1.1. In bitcoin, a cryptographic puzzle is also known as a mining puzzle. There are two categories of consensus algorithms: (i) proof-based consensus algorithms and (ii) voting-based consensus algorithms. See Fig. 4. In proof-based consensus algorithms [16], miners must give proof to update the blockchain ledger. In contrast, in voting-based consensus algorithms [17], the consensus participants vote to update the ledger. Here, miners are nodes that solve the cryptographic puzzle. The proof-based consensus algorithms are generally suitable for permissionless blockchains, whereas voting-based consensus algorithms are suitable for permissioned blockchains [16]. The type of blockchain is discussed in Section 1.4. The different consensus algorithms which fall under these two categorizations will be discussed in Section 6.

1.3. Consensus protocol protects against different attacks

The consensus protocols protect against various attacks. The three main attacks are Sybil attacks, double spending attacks, and byzantine faults. Double Spending Attacks: In the bitcoin network, nodes try to spend the same coin more than once. This attack is possible when over fifty percent of the blockchain network's processing power is under the hands of the attacker, known as a 51% attack. Consensus protocol prevents double spending attacks by authenticating each transaction by multiple network nodes. The consensus protocols also include methods to detect and reject conflicting transactions, such as Bitcoin's longest chain rule [18,19]. Sybil Attacks: When an attacker creates multiple fake nodes to manipulate or gain control of the network is known as a Sybil attack [20]. The consensus protocol protects the network against Sybil attacks. Proof-of-work is a consensus protocol which is a complex cryptographic puzzle. This cryptographic puzzle needs high computational power. The answer to this complex cryptographic puzzle is too expensive [21]. Therefore, proof-of-work makes Sybil attacks difficult for an attacker to create multiple fake nodes in order to append invalid blocks. There are other consensus protocols like algorand and proof-of-stake. In proof-of-stake, the attacker needs most of the stake in the network to create fake participants that are not feasible. Algorand assigns weight based on the node's stake in the network to prevent Sybil attacks. Byzantine Faults: The nodes start behaving maliciously in the blockchain network to manipulate the transactions for their profit [22]. This malicious behavior is known as byzantine faults. This malicious behavior includes things like nodes that initiate invalid transactions or try to spend the same transaction twice or send incorrect messages to other nodes in the network. The PBFT is a consensus mechanism that can bear at most one-third of the nodes to be faulty. The WBFT consensus protocol boosts the byzantine fault tolerance rate over time. Therefore, WBFT reduces the number of byzantine participants participating in the consensus process after each round. Here, byzantine nodes mean malicious nodes [23,24].

1.4. Types of blockchain

Blockchains are divided into two categories: permissioned and permissionless. The public blockchain is another name for the permissionless blockchain. Fig. 3 depicts the two types of permissioned blockchain: (i) consortium and (ii) private [25]. The public blockchain allows any node to take part in the consensus process. One can join and leave without permission. In a public blockchain, nodes do not know the identities of each other. Blockchain information is accessible to everyone in a public blockchain. Any node can become a miner. Bitcoin is one example of a public blockchain. Only approved nodes can join the network and participate in consensus on the private blockchain, and nodes know the identities of each other. Blockchain information is only accessible to authorized nodes, not open to everyone. The private blockchain is scalable in terms of the number of transactions processed in the network per second but not the number of nodes [26]. The enrollments in a consortium blockchain are confined to the consortium members as determined by a predetermined list of members. Nodes in consortium blockchain know

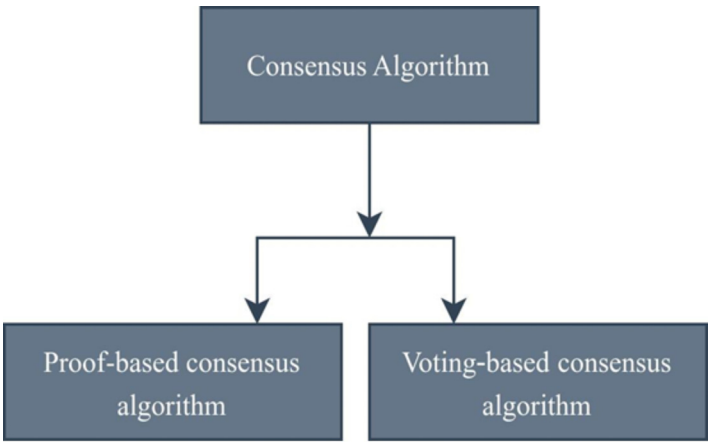

Fig. 4. Categorization of Consensus algorithms.

Table 1
Permissioned vs Permissionless Blockchain.

	Permissioned	Permissionless
Control	Consortium / Private	Public
Access	Authorized access	Free access
Node Identity Management	Known	Anonymous
Transparency	Closed	Open
Scalability (no. of nodes)	Limited (tens hundreds)	Excellent (thousands of nodes)
Network synchrony assumptions	Synchronous	Asynchronous
Throughput (transactions per second)	High (tens of thousands of transactions per second)	Limited (due to possible of chain forks)
Latency	Excellent (equal to network latency)	High latency
Blockchain application examples	Supply chain management, business contract	Smart contract, a decentralized application (DApp)

the identities of each other. A distributed ledger is only accessible to the members of a blockchain. TradeLens is an example of consortium blockchain for supply chain management [27]. The comparison between permissionless and permissioned blockchains is shown in Table 1.

The remaining paper is organized as follows. First, related work is discussed in Section 2. Second, the working of a bitcoin network is presented in Section 3. After that, scalability issues in blockchain and the three methods to address the scalability problem in blockchain are presented in Section 4. Section 5 covers industries blockchain applications. Different consensus algorithms are discussed in Section 6. The comparison of different consensus algorithms based on specific parameters has been explained in Section 7. Section 8 covers challenges and future directions. Finally, Section 9 discusses the conclusion of the paper.

2. Related work

Table 2 has done a comparison among existing surveys on scaling consensus algorithms. The table also highlights how our survey is different from the existing survey. The table compares our work with the existing surveys based on the following parameters: on-chain scalability solutions, off-chain scalability solutions, consensus algorithm scalability solutions, blockchain applications in various industries, and recent consensus algorithms like PoEWAL and WBFT. The PoEWAL was published in the year 2020, whereas WBFT was published in the year 2022.

The survey by Salimitari et al. [28] provides a comprehensive overview of the various consensus methods for resource-constrained IoT networks and highlights the challenges and opportunities in this field. The survey focuses on the challenges, particularly in terms of the limited resources available to the devices in these networks. The paper also highlights the security challenges that arise in these networks, such as the risk of double-spending and the need for secure communication channels between devices.

The survey by Xiao et al. [29] provides an overview and analysis of various distributed consensus protocols used in blockchain networks. The paper highlights the benefits and drawbacks of each consensus mechanism, such as the energy efficiency of PoS compared to PoW, the high-performance benefits of DPoS, and the security concerns associated with some consensus mechanisms. In the rapidly evolving field of blockchain and consensus protocols, several new developments have taken place since then that have not been covered in this paper. This paper has partially covered scalability solutions to address the scalability issue in the blockchain. The paper has not covered recent consensus algorithms like PoEWAL and WBFT.

The survey by Bamakan et al. [30] provides a comprehensive overview of the current state of evaluation criteria and identifies the most common evaluation metrics used in the literature. However, the paper does not compare the evaluation criteria or cover recent advancements in the field. The paper has not covered on-chain and off-chain scalability solutions at all, as well as blockchain applications, and also not covered PoEWAL and WBFT.

Wang et al. [31] focus on the consensus algorithm, a crucial component of Blockchain technology. The author presents a comprehensive consensus algorithm process model that can be applied to chain-based and directed acyclic graph (DAG) structured Blockchains. The author categorizes consensus algorithms based on their fit within the proposed process model. Finally, the author recommends choosing the appropriate consensus algorithm for different Blockchain applications. Hasan et al. [32] emphasize the concept of blockchain scalability as a multifaceted idea that can encompass various aspects, such as increasing the number of network participants or improving the capabilities of existing participants to reduce scalability limitations. It also provides a comprehensive examination of ongoing efforts to achieve blockchain scalability, including innovative techniques and mechanisms to enhance scalability, approaches for creating scalable applications using blockchain technology, and evaluations of various blockchain solutions in terms of scalability. By analyzing existing literature, the paper identifies key contributions and remaining challenges that can aid the research community in advancing their understanding of this field.

3. Working of bitcoin network

In Bitcoin, nodes initiate new transactions, and subsequently, all nodes receive these transactions. Each miner compiles the transactions into a block and solves a cryptographic puzzle. Once a miner finds a solution, they broadcast the block to all nodes. If the block contains valid transactions, nodes accept it. The block is then appended to each node's ledger, and the ledger is shared with its peer nodes. In case of invalid transactions, nodes reject the block. Miners express their acceptance by mining the next block right after the accepted one onto the blockchain [33,34]. In the bitcoin network, miners fight against each other to solve

Table 2
Comparison of our survey with existing surveys on consensus algorithms.

Reference Year	Covered Period	On chain Scalability Solution	Off-chain scalability solution	Consensus algorithms scalability solutions	Blockchain Applications in various industries	WBFT	PoEWAL
Salimitari et al. [28], 2020	2003–2019	✗	✗	√	Only IoT	✗	✗
Xiao et al. [29], 2020	1978–2019	√partially covered	√partially covered	√	√	✗	✗
Bamakan et al. [30], 2020	1998–2020	✗	✗	√	✗	✗	✗
Wang et al. [31],2021	1982–2020	✗	✗	√	✗	✗	✗
Hasan et al. [32], 2022	2007–2020	√	√	√	√	✗	✗
Our Work	1990–2022	√	√	√	√	√	√

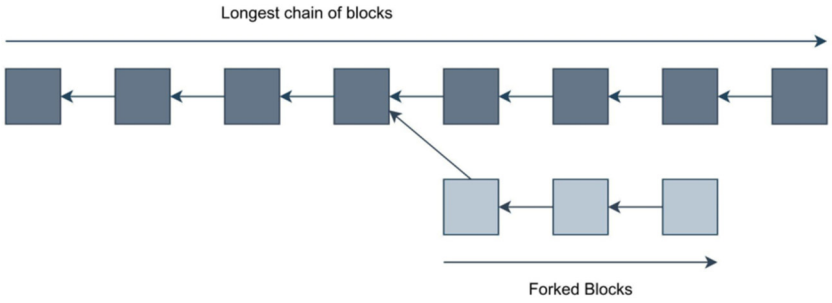

Fig. 5. Creation of forked blocks outside the main chain (longest chain).

the cryptographic puzzle. Assume two miners solve at the same instant. Now both miners append the block to their ledger and broadcast their blockchain ledger to all nodes. Nodes accept that blockchain comes from most of its peer nodes. If a node gets an equal number of two different blockchains from its peer nodes, then that node first checks the timestamp of the latest block and accepts that block with the blockchain created earlier. If the timestamp is equal, a node can accept any blockchain as per its will. Nodes accept the longest chain of blocks for synchronizing nodes in blockchain networks [35]. Blocks that are created at the same time are appended on top of the same block in the blockchain. Here, the assumption is that the networks are already synchronized with the same blockchain ledger before the creation of the blocks. This results in the formation of forks in the blockchain. The forked block is shown in Fig. 5. The blocks not part of the main chain are orphaned or pruned blocks. To solve the cryptographic puzzle, miners need to do high computation, which ultimately requires high CPU power, which consumes a lot of electricity [36]. Due to the creation of forks, miners who mined the forked block are at a loss because the miners have spent considerable energy mining the forked blocks. The network considers the longest blockchain as the main chain. Therefore, the transactions included in the forked block not present in the main chain are considered in the following block to be mined. The problem of forks can be solved by putting two restrictions on the network: the first one is that the block is mined at an interval of 10 min. The second is that the network participants ensure that their transactions remain on the main chain (longest chain) forever if six blocks are mined on top of that transactions. Consequently, it takes the bitcoin network around an hour to confirm a transaction and record it permanently on the blockchain [37]. Bitcoins are given as rewards to miners for mining new blocks. The reward consists of a transaction fee paid by the node that started the transaction and some new bitcoins that did not exist before. Every time a miner creates a new block, the network creates a certain number of new bitcoins. To prevent inflation and make bitcoin a valued asset similar to gold, the total amount of bitcoins on the network is set at 21 million [38]. Every 210,000 blocks, or about every four years, the amount of bitcoins created every block drops by 50%. Section 4 covers proof of work in detail.

4. Scalability issues in blockchain

The parameters to address scalability issues in the blockchain are shown in Fig. 6. The parameters are throughput, latency, block generation rate, and storage scalability. In bitcoin, blocks are generated every 10 min, called the block generation rate (BGR), and each block is 1 MB in size. The BGR and the block size limit the bitcoin network's throughput, that is, TPS. A transaction on the bitcoin network takes about an hour to confirm. Therefore, the bitcoin network has high latency. One can think of increasing the block size to solve the problem of low throughput. However, larger blocks take a higher time to propagate in the network. As a result of which, the latency becomes high. If the throughput and latency are improved by reducing the block generation rate, forks in the network are more likely to occur [39]. The bitcoin ledger is more than 280 GB. Therefore, a node must first download vast amounts of data while joining the bitcoin network to get a full view of the blockchain ledger. Therefore, the node joining the bitcoin network requires a large storage capacity. Hence, storage scalability needs to be addressed. Storage scalability is important for various blockchain applications, for example, blockchain-based resource-constrained IoT devices. The resource-constrained IoT devices have limited storage space. The three methods to address the blockchain's scalability issue are off-chain solutions, on-chain solutions, and consensus algorithms [32].

4.1. On-chain solutions

The on-chain methods try to increase blockchain scalability by changing a blockchain's internal characteristics. The transaction or message size can be optimized, or the network latency can be improved. The following is a list of some on-chain solutions:

Blockchain Pipelining: McConaghy et al. [40] introduced the concept of blockchain pipelining, which involves breaking down the validation and processing of blockchain transactions into multiple stages. Each stage is processed in parallel, dividing the transaction processing pipeline into smaller, more manageable components. This enables multiple nodes in the network to execute these components simultaneously. The key benefit is an increased transaction processing capacity within a given timeframe, in contrast to traditional blockchain systems where transactions are processed sequentially.

Blockchain Delivery Network: A proposed solution to scalability challenges in blockchain systems is the concept of a blockchain delivery network. The fundamental idea is to establish a network of nodes capable of efficiently handling and transmitting blockchain transactions, resulting in faster and more streamlined transaction processing compared to conventional blockchain systems [30]. In a blockchain delivery network, transactions undergo segmentation into smaller chunks, which are then transmitted among nodes within the network. Each node bears the responsibility of validating and processing a segment of the transaction, enabling parallel processing and enhancing scalability. The primary advantage of a blockchain delivery network lies in its ability to expedite transaction processing. By breaking transactions into smaller portions and concurrently processing them across multiple nodes, the network can decrease the time required for transaction confirmation. Additionally, this approach helps alleviate the load on individual nodes, contributing to enhanced system performance and reliability [32]. Beyond scalability improvements, blockchain delivery networks offer added security and privacy benefits. The division of transactions into smaller segments and their parallel processing make it more challenging for malicious actors to compromise the system, as attacking multiple nodes simultaneously becomes more complex.

Block Size Adjustment: Block size adjustment serves as a mechanism to tackle scalability challenges in blockchain systems. In a conventional blockchain, each block has a fixed size, leading to limitations on the number of transactions that can be processed within a block. This limitation can create a bottleneck within the system, constraining the overall transaction processing capacity [32]. To overcome this challenge, certain blockchain systems have incorporated block size adjustment mechanisms, allowing for dynamic modification of each block's size based on the current network conditions. The concept behind block size ad-

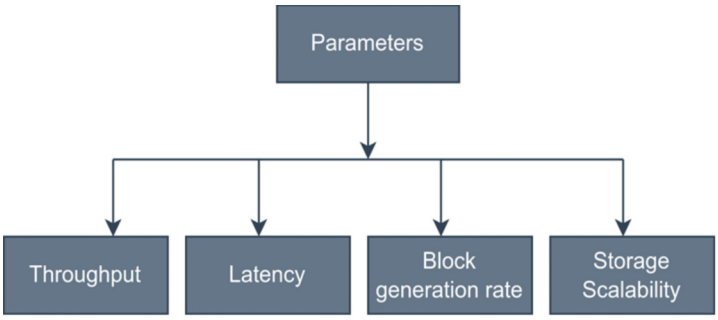

Fig. 6. Parameters to address scalability issues in the blockchain.

justment is to expand block sizes during network congestion and shrink them during less crowded periods. This adaptive approach ensures the system's efficiency and effectiveness, accommodating the increasing volume of transactions processed by the network over time [41]. There are different approaches to block size adjustment, and the specific mechanism used can vary depending on the particular blockchain system. For example, some systems may allow the block size to be adjusted by the network consensus, while others may rely on the individual nodes in the network to adjust the size of blocks based on local conditions.

4.2. Off-chain solutions

Off-chain solutions refer to methods for improving the scalability of blockchain technology by taking transactions and data storage off the main blockchain network. Here are some standard offchain solutions: Payment Channel Networks: In a payment channel network, two or more parties can open a payment channel between themselves, allowing them to transact directly without having to broadcast each transaction to the entire network. This off-chain transaction can be completed quickly and efficiently, without the need for confirmation from the whole network, and can be done with a low fee. Once the channel is closed, the final state of the channel is recorded on the blockchain, ensuring that the transactions are secure and auditable [32]. The key advantage of payment channel networks is that they allow for faster and more efficient transactions, as users can transact directly with each other without having to broadcast each transaction to the entire network. This can reduce the network load, improving the system's overall performance and scalability [42]. In addition to improving scalability, payment channel networks can provide additional benefits, such as increased privacy and reduced fees. For example, by transacting off-chain, users can keep their transactions private and avoid paying high fees to broadcast each transaction to the network. Sharding: Sharding presents a proposed solution to address scalability challenges in blockchain systems. The fundamental concept behind sharding involves dividing a blockchain network into smaller and more manageable entities, known as "shards." Each shard assumes the responsibility of processing a specific portion of the transactions within the network, facilitating parallel transaction processing and enhancing scalability [32].

Each shard operates autonomously, managing its state and transactions in a sharded blockchain network. This approach aids in alleviating the burden on individual nodes within the network, as it reduces the number of transactions each node needs to process. Consequently, this can lead to quicker and more efficient transaction processing, ultimately enhancing the scalability and security of the network [41]. There are different approaches to sharding in blockchain systems, and the specific mechanism used can vary depending on the blockchain. For example, some blockchain systems may implement random sharding, where transactions are randomly assigned to different shards. In contrast, others may implement transaction-based sharding, where transactions are assigned to shards based on their content or other characteristics [43]. Overall, Sharding is a crucial tool for tackling scalability challenges in blockchain systems and stands as an area of active research and development. By partitioning the network into smaller, more manageable components, blockchain systems can sustain effectiveness and efficiency as the volume of processed transactions increases over time. However, it is essential to acknowledge that sharding introduces new challenges, including the need to maintain consistency and data integrity across the network. Further efforts are required to comprehensively grasp and unlock the potential benefits that sharding can bring to blockchain systems [44].

4.3. Consensus algorithm solutions

Nodes in distributed systems reach a consensus on a single value through consensus algorithms. The consensus algorithm improves scalability in blockchain systems by reaching consensus faster on a single value. Therefore, different consensus algorithms have been developed to improve scalability by decreasing the time gap between block creation [31]. The proof-of-work consensus algorithm is an excessive computation algorithm requiring considerable time to find the cryptographic puzzle. This delay in the confirmation of the transaction directly affects latency and throughput. The proof-of-work has low throughput and high latency. Alternative consensus algorithms have been designed to find the problem of scalability in proof-of-work. The various algorithms are proof-of-burn (PoB), proof-of-stake (PoS), proof-of-burn (PoB), proof-of-elapsed time (PoET), Bitcoin-NG, ByzCoin, algorand, and delegated proof-of-stake (DPoS). The energy-efficient algorithms are PoS, PoB, PoET, and DPoS. Moreover, the bitcoin-NG, byzCoin, and algorand subsequently improve the scalability of cryptocurrency applications. These algorithms are suitable for permissionless blockchains [45]. The PoEWAL has been designed for IoT-based blockchain applications. The PoEWAL is designed for resource-constrained IoT devices. The PoEWAL is also suitable for permissionless blockchains [46]. The Raft, Practical Byzantine Fault Tolerance (PBFT), and Weighted Byzantine Fault Tolerance (WBFT) [47] have been specifically designed for permissioned blockchains. These algorithms exhibit low latency and high throughput, outperforming traditional consensus algorithms like Proof-of-Work. However, there is room for improvement in terms of scalability, especially concerning the number of nodes. The different consensus algorithms have been discussed in more detail in Section 4.

5. Application of blockchain technology in the industry

Blockchain technology holds the potential to revolutionize various industries during the Fourth Industrial Revolution (Industry 4.0). Below are some key applications of blockchain technology in Industry 4.0:

5.1. Intelligent supply chain

The Internet of Things (IoT) has transformed supply chain management. Due to limited computing and storage resources in IoT, logistics data is traditionally stored in centralized cloud centers. However, these centralized cloud centers are susceptible to data manipulation and represent a single point of failure [48]. The integration of blockchain technol-

ogy into supply chain management offers a secure, decentralized, and transparent approach to tracking goods and materials throughout the supply chain. Blockchain enables all parties in the supply chain to access a shared, tamper-proof record of transactions and inventory levels [49]. This enhances transparency and trust among supply chain partners, reduces the risk of fraud and errors, and facilitates faster and more efficient supply chain operations. Additionally, smart contacts can automate specific supply chain processes, such as triggering payments when goods are delivered. Blockchain technology can help increase efficiency, reduce costs, and improve the transparency and trust of supply chain transactions [50].

5.2. Innovative healthcare

Blockchain technology can securely and transparently track the movement of medical supplies, drugs, and other pharmaceuticals through the supply chain, thereby diminishing the risk of counterfeit drugs and enhancing the overall safety and efficacy of the healthcare system [51–53]. Smart contracts can also automate specific healthcare processes, including claims processing, appointment scheduling, and medical research. By doing so, smart contracts contribute to reducing administrative burdens for healthcare providers and improving the overall efficiency of the healthcare system [54].

5.3. Intelligent transportation

Using blockchain, transportation companies, and logistics providers can access a shared tamper-proof record of transactions and inventory levels [55]. Blockchain can improve transparency and trust among supply chain partners, reduce the risk of fraud and errors, and enable faster and more efficient transportation operations. Additionally, blockchain technology can securely and transparently track the movement of vehicles and people and can help improve overall safety and efficiency. For example, blockchain can track the location and condition of vehicles and automatically pay tolls or other charges based on usage [56]. Smart contracts can also automate transportation processes, such as cargo tracking, vehicle maintenance, and logistics planning. Smart contracts can help to reduce administrative burdens for transportation companies and improve the overall efficiency of the transportation system [57].

5.4. Intelligent agriculture

Using blockchain, farmers, agribusinesses, and logistics providers can access a shared, tamper-proof record of transactions and inventory levels, improving transparency and trust among supply chain partners, reducing the risk of fraud, and enabling faster and more efficient agricultural operations [58,59]. Additionally, blockchain technology can securely and transparently track the movement of agricultural products from farm to table. It can help improve the food supply chain's overall safety and efficiency [60]. Blockchain can include tracking food products' origin, quality, and authenticity and ensuring compliance with food safety regulations. Smart contracts can also automate certain agricultural processes, such as crop planting and harvesting livestock tracking. Smart contracts can help to reduce administrative burdens for farmers and improve the overall efficiency of the agricultural system [61]. Overall, blockchain technology has the potential to make agriculture more efficient, secure, and transparent by creating an immutable record of all transactions, ownership, and quality of the product. This way, stakeholders can have a trustable source of information which can help reduce the risk of fraud and errors.

5.5. Intelligent grid

Blockchain technology has the potential to transform the way the smart grid operates by providing a secure and decentralized way to manage and track the flow of electricity [62]. By using blockchain, the smart grid could be made more efficient, transparent, and resilient. One example would be using blockchain to track the distribution and consumption of renewable energy, allowing for more accurate tracking and settlement of renewable energy credits. Additionally, blockchain could be used for secure communication and data sharing between grid participants and for managing distributed energy resources such as storage and generation. Blockchain technology can improve efficiency, security, and reliability [63]. Additionally, blockchain can help integrate renewable energy sources into the grid by enabling the creation of peer-to-peer energy trading platforms. Energy trading platforms allow prosumers (consumers who are also energy producers) to sell their excess energy to their neighbors, creating a more decentralized and efficient energy market [64]. In summary, blockchain technology can improve energy transactions' efficiency, security, and transparency in intelligent grids by automating trades, providing tamper-proof record-keeping, and enabling peer-to-peer energy trading.

6. Different consensus algorithms

Consensus algorithms emerge in the theory of replicated state machines [65–67]. A replicated log is used to implement replicated state machines, as shown in Fig. 7. Each server consists of a state machine, a log, and a consensus module. On each server, there is a log that holds a series of commands. This log is kept identical across all servers, ensuring that the sequence of commands contained in the log is the same for each server. Consequently, each state machine runs an identical sequence of commands. This section has discussed various consensus algorithms, starting with proof-based algorithms and followed by voting-based algorithms. In Section 6, a comparison of different consensus algorithms has been conducted based on various parameters.

6.1. Proof-based consensus algorithms

The proof-based consensus algorithms include PoW, PoS, DPoS, PoB, PoET, Bitcoin-NG, ByzCoin, and PoEWAL. In these algorithms, each participant can cast a vote on the decision or proposal under consideration. The voting process can be structured in various ways, depending on the specific needs and requirements of the system. A common approach is a simple majority vote, where a decision is reached if more than half of the participants agree.

6.1.1. Proof-of-work (PoW)

Dwork and Naor [68] introduced the concept of Proof-of-Work (PoW) as a solution to combat spam emails. The idea is that a certain amount of work must be done in order to send a valid email, which would discourage attackers from sending spam emails in the future. The work must be moderately difficult but doable for the sender. The work must be simple for the receiver to check. In the Hashcash [69] proof-of-work system, miners must demonstrate that they have completed a certain amount of work before they can propose a new block. The attacker would be discouraged from proposing new blocks or altering existing ones. The bitcoin PoW algorithm ensures consensus in a decentralized setting using challenge-response. Consensus is reaching a joint agreement in a distributed or decentralized multi-agent system. Under the challenge-response protocol, the network issues a challenge that miners within the network aim to solve. The block is granted to the miner, who is first able to demonstrate that they have successfully solved the challenge [70]. A mining puzzle's difficulty is dynamically changed so that blocks are generated on average once every ten minutes (the block hash contains a specific number of leading zero bits) [1]. One of the two crucial characteristics of bitcoin is block frequency, defined as the latency of 10 min(per block). The other crucial parameter is block size, which is set at 1 MB. The parameters "block frequency" and "block size" set a limit on the number of transactions per second (TPS) that can be processed. The Bitcoin network has an average TPS capacity of 7, significantly lower than PayPal's 500 TPS and VISA's 4000 TPS. To enhance

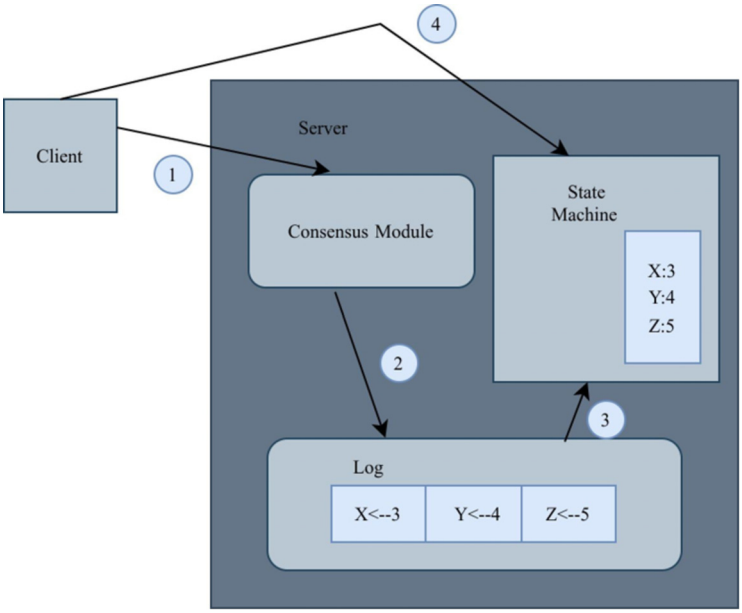

Fig. 7. State Machine Replication.

the Bitcoin network's throughput, a straightforward solution is to increase the block size. However, this approach introduces challenges, as larger blocks take longer to propagate through the network, increasing the risk of forks and double-spending attacks [71–73]. To maintain the liveliness and security features of Bitcoin, forks must occur relatively infrequently, typically around once every 60 blocks. Reducing the block frequency can address the issue of forks and double spending but may lead to network instability with disagreements among nodes. As a general principle [1], a block is permanently committed to the blockchain once around six new blocks have been mined on top of it. To ensure that a transaction remains on an authoritative chain, a client has to wait about an hour (a block is mined at an interval of 10 min, so for six blocks, total latency of 60 min). The PoW consensus algorithm is the "Probabilistic Algorithm" [74]. In addition to these high latency and low throughput, the PoW algorithm wastes a lot of electricity, near about 0.1-10 GW, which was roughly estimated in 2014 [75]. Because of the poor performance of PoW, the PoW consensus algorithm cannot be applied in applications that require low latency and high throughput. Therefore, different consensus algorithms are designed for different applications.

6.1.2. *Proof-of-stake (PoS)*

In proof-of-work, miners compete to mine the next block. In contrast, PoS selects a node based on its proportional stake in the network, representing the amount of cryptocurrency held by the node. The chosen node doesn't compute a hash (target value); instead, it utilizes a digital signature to demonstrate its ownership stake. From the first day, all coins are there in the network. There is a transaction fee to compensate the miners [76]. Mining rewards and coin creation are not concepts in PoS. Although this method removes high computation of proof-of-work, two new problems come into the picture. First, proof of stake depends on the node's stake (the highest amount of cryptocurrency a node holds) for the selection of a miner. Therefore, the blockchain network using proof-of-stake becomes centralized to some extent. Second, a problem called "nothing at stake" exists in proof of stake [29,70]. This problem leads to a situation where the selected node cannot be punished for misbehaving. For example, nothing can stop the selected node from proposing two new blocks to gain more transaction fee rewards. However, the proof of stake has undergone modifications to address these problems. Furthermore, proof-of-stake and its variants, such as DPoS, rely on financial principles (stakes) and are, therefore, not suitable for applications such as the Internet of Things (IoT) and Big Data [28,30].

6.1.3. *Delegated proof-of-work (DPoS)*

Delegated proof-of-stake is comparable to proof-of-stake. Unlike direct democracy, delegated proof-of-stake operates as a representative democracy [77], where all stakeholders vote to select specific nodes as witnesses and delegates. These chosen witnesses are tasked with creating new blocks and are rewarded for their contributions. The delegates play a role in maintaining the network and suggesting changes, such as reward amounts, block sizes, or transaction fees. The network chooses N witnesses with the highest voting in each election. The number N is decided such that voting stakeholders equal to or greater than 50 percent believe there is sufficient decentralization. For decentralization, each stakeholder votes for at least as many witnesses as they choose. For voting and spotting a dishonest witness or delegate, DPoS has an internal mechanism [76,78,79]. Delegated proof of stake (DPoS), used by the cryptocurrency bitshares, provides better throughput and latency than proof of stake (PoS) but runs the danger of centralizing the blockchain network. Around 100k transactions may be processed per second using a delegated proof of work (TPS). Additionally, it takes a maximum of 3 s, or 1.5 s on average, to attach a block to the blockchain [28,31,70].

6.1.4. *Proof-of-burn (PoB)*

To demonstrate their work, miners burn coins in a process known as proof of burn. In this case, burning entails sending a small amount of money to an unusable address. Depending on how much money a miner has burned, he or she gets chosen to find the next block. Proof of burn (PoB) does not use energy or resources like proof of work (PoW). A user needs to be willing to lose some money upfront to reap the rewards down the road [28,30,70]. The cryptocurrency sent to an unspendable address cannot be utilized again by anyone as this type of address is generated randomly and has no association with a private key. No private key with this address means nobody can access and spend the cryptocurrency stored at that address. To reach a consensus, proof of work cryptocurrencies like bitcoins is burnt by proof of burn algorithms. A cryptocurrency called slimcoin (SLM) burns bitcoins as a mining method [80]. More cryptocurrency a user burns, the more opportunities he/she will get to solve the block. Like PoS, the wealthy are likely to become wealthier in this method. If one day, the value of the coins burned in PoW becomes less than the value in PoB, then PoW will be less energy efficient than PoB.

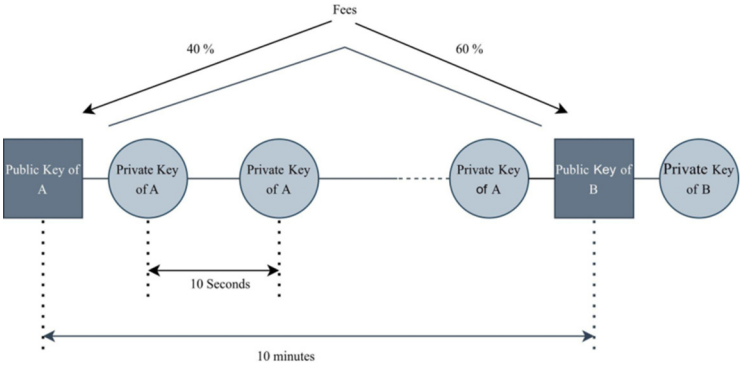

Fig. 8. Structure of Bitcoin-NG chain. 40% of the fee is given to the leader, and the next leader gets 60%.

6.1.5. Proof-of-elapsed time (PoET)

Proof of Elapsed Time (PoET) is a consensus algorithm proposed by Intel that functions similarly to PoW but with lower energy consumption. In PoET, miners are required to solve a hash problem similar to PoW. However, the winning miner is selected based on a randomly generated waiting time rather than a competition to solve the next block. The miner whose timer expires first is chosen as the winner. The accuracy of the timer execution is verified using a trusted execution environment such as Intel's Software Guard Extension (SGX) [81]. This approach aims to reduce the energy footprint associated with traditional PoW while ensuring fairness in block selection through a randomized waiting time mechanism. One of the benefits of PoET is its compatibility with the Internet of Things (IoT) due to its reduced computational demands [82]. However, a significant drawback of PoET is its dependency on Intel, which goes against the decentralized philosophy of blockchain technology [83].

6.1.6. Bitcoin-NG

Eyal et al. [84] proposed a scalable consensus protocol called Bitcoin-NG. Bitcoin-NG addresses the two major problems in PoW: high latency and low throughput due to consensus finality. In Bitcoin-NG, the network's propagation delay solely determines the latency. This improved performance is achieved by decoupling the blockchain operation into two planes: transaction serialization and leader election. The protocol divides time into epochs, and traditional bitcoin PoW is used to elect a single leader in each epoch for serializing transactions until the epoch concludes [31]. The Bitcoin-NG introduces two types of blocks. The first block is a key block for leader election, and the other is a microblocks for ledger entries. Like bitcoin, key blocks are mined by miners using the PoW protocol. The node that mined the key block first becomes the leader. The desired hash value (a specific amount of zeros bits at the prefix or a goal value), a nonce value, the current Unix time, the preceding block hash (which can be either of a key block or a microblock, generally the latter), and a coinbase transaction are all contained in the keyblock (arbitrary bits). To maintain the average generation rate between two successive keyblocks, the target value based on Unix time varies deterministically to modify the difficulty (target value). The nodes choose the longest or heaviest chain to reduce the fork, just like in bitcoin [84]. Initially, the leader is the node that mined the keyblock first. The leader produces the microblocks at a fixed rate of less than a predetermined maximum. The size of each microblock is constrained to a predetermined maximum. When a microblock's timestamp is in the future, the microblock is deemed invalid. A microblock comprises the Merkle root of its ledger entries, the Merkle hash of the previous block, the current Unix time, and the private key corresponding to the public key of the most recent keyblock [84]. A microblock is valid if its ledger entities are valid as per the specification of the state machine and contain a valid signature (private key). Fig. 8 depicts the structure of Bitcoin-NG. One important point to note is that the leader can generate microblocks cheaply and quickly because the leader does not require mining to generate microblocks [84]. The new leader may not have heard all the microblocks of the previous leader. This is the common situation on leader switching if there is a frequent generation of microblocks. To prevent the microblock from being pruned by the new block, a node must wait for the network's propagation delay before taking the microblock into account in the chain [84]. A leader is motivated to mine by rewarding

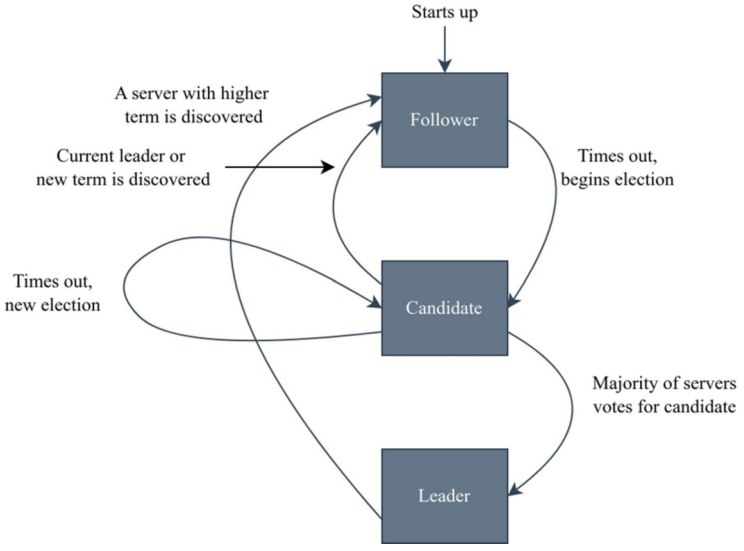

Fig. 9. Transitions of server states.

her computation by the protocol. This reward consists of two parts. One part is that set amount given to the key block's generator. The other part is a fee for each ledger entry. This fee is split between the leader who included that specific ledger entry in the microblocks and the next leader who generates the next block. Forty percent of the fee is allocated to the current leader, while the subsequent leader receives 60% of the fee, as depicted in Fig. 9 [84].

6.1.7. ByzCoin

ByzCoin, a Byzantine consensus mechanism proposed by Kogias et al. [85], commits transactions within one minute of submission. The use of communication trees to enhance transaction commitment and verification during regular operation reduces this delay to 30 s. ByzCoin generates jointly signed transaction blocks within one minute of submitting a transaction, preventing selfish mining and double-spending attacks. It currently achieves higher throughput than PayPal, with a confirmation latency of 15 to 20 s [28].

In ByzCoin, a consensus group utilizes a collective signing protocol (CoSi) [86] for prompt transaction commitment. A proof-of-membership mechanism based on proof-of-work is employed to join a consensus group. Miners receive a share of the consensus group when they discover a new block, and this share serves as proof of their membership in the group of trustees. Similar to Bitcoin-NG, ByzCoin has two types of blocks: Keyblocks and Microblocks. The current leader generates a new microblock every few seconds, using the CoSi-based PBFT protocol to commit and collectively sign the transactions. Key blocks in ByzCoin function similarly to Bitcoin-NG, electing a leader and creating shares for miners to prove their membership in ByzCoin's consensus group [85]. Each microblock contains a set of transactions, a collective signature, and hashes (hashes of the previous microblock and keyblock). To ensure total ordering, the hash of the preceding microblock is utilized, and the hash of the keyblock shows which consensus group and leader produced the microblock's signature. A new consensus group is established with every newly mined keyblock, resulting in the next era's microblocks being collectively signed by a distinct set of public keys. ByzCoin guarantees the irreversible commitment of each microblock, indicating transactions to be stored, regardless of the current leader's behavior [85].

6.1.8. Proof-of-elapsed work and luck (PoEWAL)

A lightweight consensus algorithm named "Proof of Elapsed Work and Luck (PoEWAL)" was proposed by Raghav et al. [46] for application in the Internet of Things (IoT) blockchain. Primarily designed for resource-constrained IoT devices participating in the consensus mechanism of blockchain applications, PoEWAL is energy-efficient, requiring less computational power and exhibiting low latency. In PoEWAL, the cryptographic puzzle is solved within a predetermined window of time, similar to proof of work. The miner accomplishes this task within the allotted time, providing proof of elapsed work. The cryptographic puzzle involves concatenating the hash of the most recent block with a random number, known as a nonce value. The solution to the puzzle is the hash value with the most consecutive zeros beginning with the Most Significant Bit (MSB). In case of a collision, indicating proof of luck, when two or more miners achieve the same number of consecutive zeros, the miner with the lower nonce value is declared the winner. The likelihood of multiple miners having the same lowest nonce value is minimal, and even in such cases, the algorithm compares the hash values, selecting the one with the smallest hash value as the winner [46].

6.2. Voting-based consensus algorithms

The voting-based consensus algorithms are Raft, PBFT, Algorand, and WBFT.

6.2.1. Raft

The Raft consensus algorithm [87] manages a replicated log of client-state machine commands. Initially, a server is elected as the leader, as-

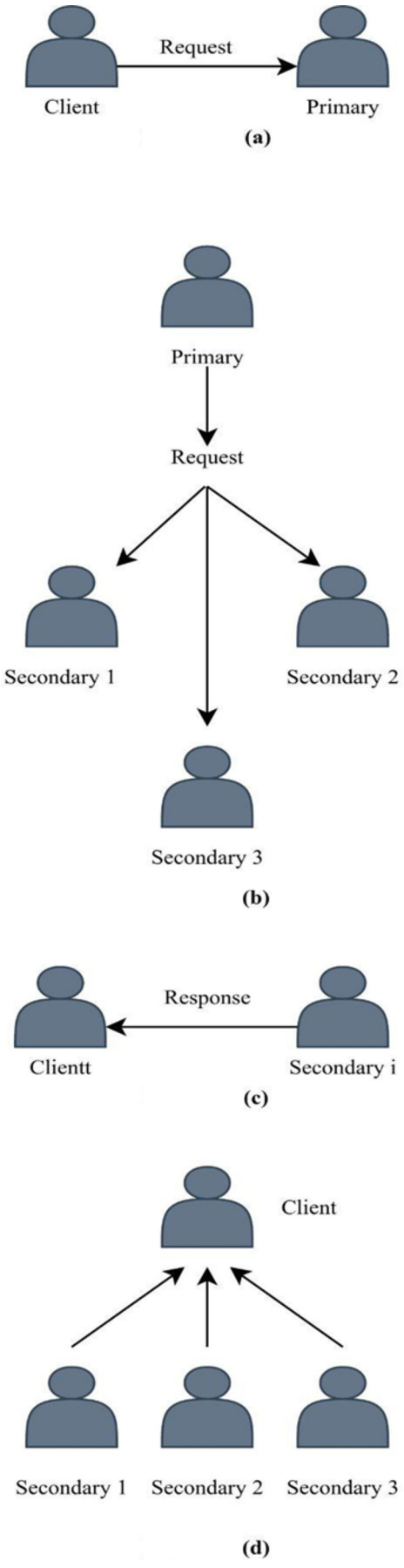

Fig. 10. The working of PBFT algorithm.

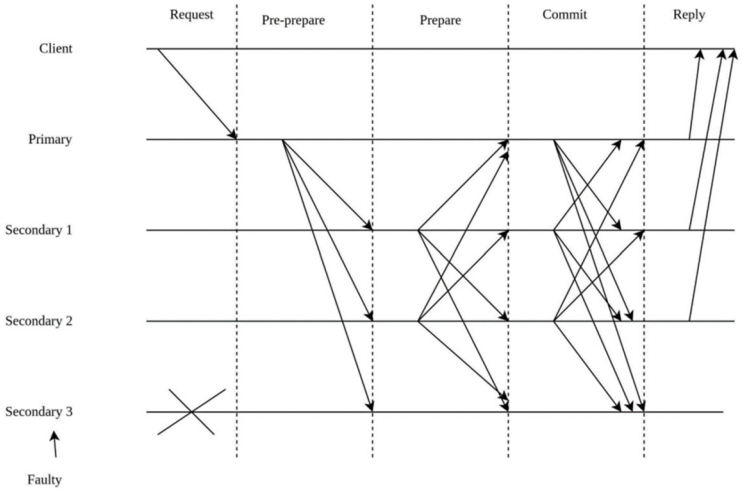

Fig. 11. Three-phase protocol of PBFT.

Table 3
Three-phase protocol of PBFT.

Three-phase	Message Form	Symbols meaning
Pre-prepare	Primary multicasts a message and gives the request a sequence number of n. The format of the message is <<PRE-PREPARE,v,n,d> σ_p, m>	v: current view number d: message digest σ_p: private key of primary m: message n: sequence number
Prepare	The backup enters the prepare phase by multicasting a message titled <Prepare,v,n,d,i> σ_i to all other replicas if it accepts the pre-prepare message.	i: replica number σ_i : private key of replica i
Commit	Multicast the message <Commit,v,n,d, i> σ_i to all replicas, including the primary.	

suming complete responsibility for overseeing the replicated log. The leader accepts log entries from clients and notifies servers when they can execute log entries. In the event of a leader's failure or disconnection, Raft initiates the election of a new leader. This algorithm can be broken down into three distinct sub-problems: leader election, log replication, and safety.

At any given time, a server can exist in one of three possible states: leader, follower, or candidate. Fig. 9 illustrates the transitions among these states. Followers are passive, responding to requests from candidates and leaders without initiating requests themselves. The leader manages all client requests, redirecting the client to the leader if initially contacted by a follower. The third state involves the election of a new leader, referred to as a candidate [28]. Every term begins with an election where one or more candidates compete for leadership positions. For the remainder of the term, the winner is in charge. If two candidates receive an equal number of votes in an election, there will be no winner. A new term and new elections will soon begin. Remote procedure calls (RPCs) are used in raft servers for communications. Two types of RPCs are required only for the basic consensus algorithm like Raft. The first one is RequestVote RPCs which are used by the candidate during elections. The other one is Append Entries RPCs which leaders use to replicate log entries and generate a heartbeat in some way. One thing to remember is that once elected, a candidate must eventually store all the committed entries from prior terms to take over as leader. This suggests that new entries to the log flow from leaders to followers and that leaders always keep current entries the same. Raft is as efficient as Paxos [88–90]. One can understand Raft more easily than Paxos because Raft's structure is different from Paxos.

6.2.2. Practical byzantine fault tolerance (PBFT)

The Byzantine Fault Tolerance (BFT) consensus technique, specifically the Practical Byzantine Fault Tolerance (PBFT) algorithm, addresses the Byzantine General problem. This algorithm relies on state machine replication across multiple nodes, requiring a minimum of 3f + 1 replicas, where "f" represents the number of faulty replicas [91]. The replicas undergo a series of configurations known as "views," with one replica designated as the primary and the others as secondary. Views are sequentially numbered, and if an issue with the primary is detected, it undergoes a change. Each view is assigned a unique number "v," and the replicas only accept messages from the active view [28,31]. "The operation of the PBFT algorithm is depicted roughly in Fig. 10, where (a) shows a client requesting the primary node to initiate a service, (b) shows the primary node broadcasting the request to the backup nodes, (c) illustrates the execution of the request by the replicas and the sending of a response back to the client, and (d) depicts the requirement for the client to receive f + 1 identical responses from various replicas to determine the output of the operation." "Fig. 11 depicts the normal operation of the PBFT algorithm. Upon receiving a client request, the primary node, p, initiates a three-phase protocol to broadcast the request to the replicas atomically. These three phases pre-prepare, prepare, and commit, are described in Table 3 and Fig. 5. For a new block to be added, the PBFT method requires a majority vote from all participating nodes and obtains consensus when a two-thirds agreement is reached, even in the presence of up to one-third of malfunctioning nodes. This method is more efficient than proof-of-work, accelerates the consensus process, and does not require assets, unlike proof of stake." Crash faults, network or partitioned faults, and Byzantine faults are the three types of faults that can occur in distributed consensus. Table 4 shows different consensus algorithms can tolerate different faults.

Table 4
Different algorithms for different faults.

Algorithm	Crash Faults	Network Faults	Byzantine Faults
PAXOS	YES	YES	NO
RAFT	YES	YES	NO
BFT	YES	YES	YES
PBFT	YES	YES	YES

Table 5
Techniques to Mitigate Three Challenges Faced by Algorand.

Challenges	Solutions
Sybil Attacks	Weighted Users: Each user is assigned a weight based on the money in their account by Algorand to prevent Sybil attacks. Consensus is ensured by Byzantine agreement until a weighted fraction (>) of the users are truthful.
Scalability (number of users)	Consensus by Committee: Byzantine agreement achieves scalability by selecting a committee (a subset of participants) from all participants for each protocol stage. Participants adhere to the protocol messages for a mutually agreed-upon block. Committee members are randomly chosen using Verifiable Random Functions (VRFs) within Byzantine agreement (BA*).
Denial of Service Attacks (targeting committee members)	Cryptographic Sortition: Users compute Verifiable Random Functions to privately determine committee membership. Selection proof and priority are provided to other users in network messages. Participant Replacement: In BA*, to counteract attacks, committee members limit messaging to one, reducing attack opportunities.

6.2.3. Algorand

A new byzantine agreement (BA*) protocol is used in Algorand to agree on the next block in the blockchain [92]. Algorand confirms new transactions with low latency and with no forks. Users in algorand compute verifiable random functions (VRFs) [83] to determine their selection for proposing a new block. Algorand faces three challenges: Sybil attacks, scalability in terms of the number of users, and denial of service attacks. These challenges are addressed using several techniques as shown in Table 5. Cryptographic Sortition is computed by all algorand users to know whether they are elected to propose a block or not. If a user is elected, then sortition provides that user with a priority and proof of the user's priority. Because sortition is a random process, a block is proposed by multiple elected users. It is the priority which tells others which block to adopt. Elected users broadcast their proposed block through the gossip protocol, along with their proof and priority. Algorand achieves the following goals: no forks, no proof-of-work, no transaction confirmation delays, trivial computation, perfect scalability, and great security.

6.2.4. Weighted byzantine fault tolerance (WBFT)

Hongwu et al. [47] introduced a Weighted Byzantine Fault Tolerance (WBFT) consensus protocol designed for consortium blockchains. This consensus mechanism incorporates a dynamic weighting system for consensus nodes to mitigate malicious behaviors. After each consensus round, nodes undergo a weight update, and those with weights exceeding a specific threshold are permitted to participate in the consensus process, thereby enhancing the security of the blockchain network. Weighted Byzantine Fault Tolerance eliminates the commit step of the Practical Byzantine Fault Tolerance algorithm for communication optimization without affecting the consensus result. Dynamic Weighting Mechanism: In WBFT, each node possesses an initial weight or base weight (w) and a running weight (wi), determining the node's activity in the consensus mechanism. The management node decides which nodes to involve in consensus based on the node's threshold value, calculated using the formula wi / w.

Nodes with higher threshold values are selected for consensus, while nodes with lower threshold values participate to a limited extent. WBFT incorporates feedback mechanisms where nodes report on the actions of nearby nodes, leading to adjustments in a node's running weight following each consensus round. Active and truthful engagement increases a node's weight, while less active or malicious behavior decreases its weight [47].

Three-Stage Communication Optimization: As depicted in Fig. 12, the primary node (p) assesses the threshold values of secondary nodes 1, 2, and 3 upon receiving a request. If a secondary node's threshold value is lower, as in the case of secondary node 3, it is excluded from the consensus procedure. The pre-prepare state, akin to the PBFT state, involves the primary node forwarding the request to secondary nodes 1 and 2. The prepare state follows, similar to PBFT, with certain nodes excluded from the consensus process. The primary node and secondary nodes 1 and 2 respond to the sender node upon entering the reply state [47]. The participation of nodes in the consensus mechanism is reduced based on their weights, excluding nodes with lower weights. This reduction minimizes malicious activities, ensuring system security and enhancing overall system performance.

Table 6
Throughput vs Node Scalability of Different Consensus Algorithms.

	<20 nodes	>1000 nodes
>1000 tx/s	PBFT	DPoS, PoET, Raft, ByzCoin
100 <tx/s <1000		Bitcoin-NG, Algorand
<100 tx/s		Standard PoW Protocols (e.g., Bitcoin), PoS, PoB

7. Comparisons of proof-of-work with different consensus algorithms

Table 6 describes the throughput vs node scalability of different consensus algorithms. Here throughput means the number of transactions validated per second in the distributed platform. The PBFT algorithm has high throughput but poor node scalability when compared with standard proof of work. The algorithms DPoS, PoET, Raft, and ByzCoin have higher throughput than PoW but node scalability is similar to PoW. And if you look at Bitcoin-NG and Algorand have medium throughput and good scalability in terms of the number of nodes. In Table 7, some parameters have been considered to evaluate the strengths and constraints of each algorithm. The parameters are consensus finality, latency, adversary tolerance, double spending attacks, and energy efficiency. Consensus finality: If a non-faulty node n attaches block k to its ledger before attaching block k' then no non-faulty node attaches block k' before block k to its ledger [93]. Latency: Latency is defined as the time duration from when a transaction is broadcasted to the blockchain network, until when a consensus is agreed upon on the proposed transactions [94,95].

Double Spending attacks: When an attacker tries to use a particular amount twice. This is possible when an attacker does a transaction to add in a block and after some time, he does another conflicting transaction to add it into a new forked incorrect block, seeking to reverse the transaction he has done before [96,97]. Adversary tolerance: In PBFT, the network can still arrive at a consensus, if at most one-third of all participating nodes are faulty (faulty nodes mean nodes start behaving maliciously). In proof-of-work, the network can still reach a consensus if the adversary's calculating power is less than 25% of the total calculating power [98]. In WBFT, the byzantine fault tolerant rate is better than PBFT [47]. Energy efficiency: PoW requires high computation to find the block's hash. This high computation consumes a lot of energy (electricity). PoS, DPoS, Raft, PBFT, and Algorand are energy efficient. PoB and PoET are not energy efficient while Bitcoin-NG and ByzCoin are partially energy efficient [98]. PoEWAL has low energy consumption. Basically, this PoEWAL is a lightweight consensus algorithm designed for resource-constrained IoT devices [46]. Table 8 compares the consensus algorithm based on the decentralization level and permission model. The permission model can be categorized as permissionless and private. Permissionless means anybody can join the network, that is the nodes are not required to know the identity of other nodes. Private means only authorized participants can join the blockchain platform, that is nodes

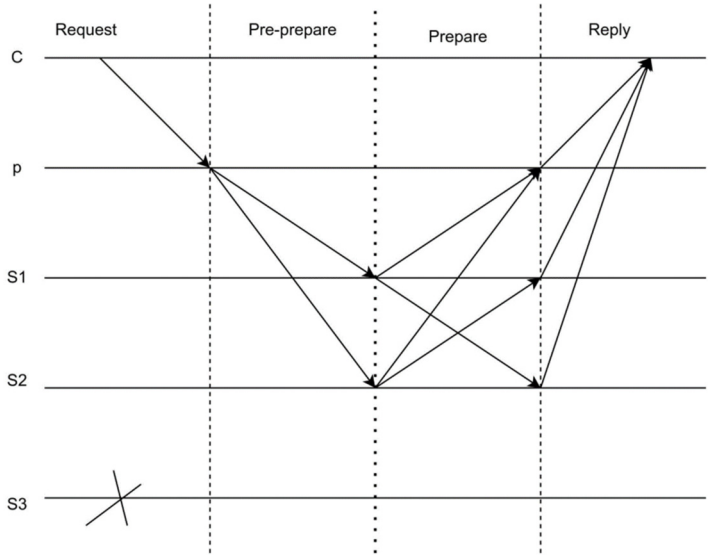

Fig. 12. The improved communication protocol of WBFT.

Table 7
Comparison of various consensus algorithms based on certain parameters.

Consensus Algorithms	Consensus finality (No forks)	Latency	Adversary tolerance	Double spending attacks	Energy Consumption
PoW	No	High	25% Computing Power	Vulnerable	High
PoS	No	Medium	<51% Stakes	Difficult	Low
DPoS	No	Medium	<51% Validators	Vulnerable	Low
PoB	No	High	<25% Computing Power	Vulnerable	High
PoET	No	Low	Unverified	Safe	High
Raft	Yes	Low	<50% Crash Fault	Safe	Low
PBFT	Yes	Low	<33% Faulty Replicas	Safe	Low
Bitcoin-NG	No	Medium	<50% Computing Power	Vulnerable	Medium
ByzCoin	No	Medium	<33% Faulty Replicas	Vulnerable	Medium
Algorand	Yes	Medium	<33% Weighted	Safe	Low
PoEWAL	Yes	Low	<25% Computing Power	Safe	Low
WBFT	Yes	Low	Byzantine Fault-Tolerant rate increases over period	Vulnerable	Low

need to know the identity of other nodes. And table 9 shows the application of different consensus algorithms and Fig. 13 shows the statistics of blockchain uses in different sectors till 2021.

8. Challenges and directions for future research

Blockchain can be applied in various domains because of the unique properties of blockchain technology. The main difficulties are how to implement blockchain technology to meet particular application requirements. Each application gives rise to different needs, and a fresh or customized implementation of blockchain technology is required. There are several challenges and future research directions for scaling consensus algorithms for blockchain technology:

Scalability: One of the main challenges in scaling consensus algorithms is increasing the number of transactions processed per second while maintaining security and decentralization. Scalability requires finding a balance among security and performance and may involve trade-offs.

Security: Another challenge is maintaining security in the face of increased scalability. As the number of transactions and nodes on the

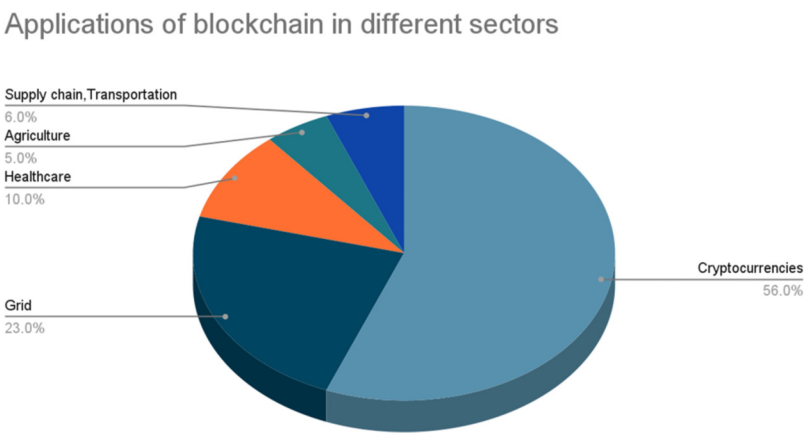

Fig. 13. The pie chart shows the percentage uses of blockchain in different sectors in 2022.

network increases, it becomes more difficult to secure the network and prevent malicious actors from gaining control.

Decentralization: Scaling consensus algorithms can also impact the decentralization of the network. Some scaling solutions, such as sharding, may lead to increased centralization as certain nodes or shards become more powerful than others.

Interoperability: Interoperability between different blockchain platforms is also a challenge. In light of the expansion of blockchain platforms, it becomes increasingly important to develop solutions that allow for cross-chain communication and interoperability.

Privacy: Another major issue when scaling consensus methods is privacy. As the number of transactions increases, it becomes more challenging to maintain the anonymity of users and protect their sensitive data.

Energy Consumption: Energy consumption is an important concern when it comes to scaling consensus algorithms, particularly with proof-of-work consensus mechanisms, solutions that aim to optimize the usage of the energy while maintaining the security of the network is an important area of research.

Hybrid Consensus: Investigating and developing hybrid consensus algorithms that combine elements of multiple consensus mechanisms, such as proof-of-work and proof-of-stake, or different sharding mechanisms.

Blockchain Scalability with AI: Combining the power of blockchain with machine learning as well as Artificial Intelligence to improve the scalability, security, and privacy of blockchain technology. Governance: Develop decentralized governance models that can handle the scaling of blockchain networks, especially for public blockchain, where a decentralized decision-making process is needed to govern the protocol, network upgrades, and other critical issues. Overall, scaling consensus algorithms for blockchain technology is a complex and ongoing research area that requires a multi-disciplinary approach, combining expertise in computer science, mathematics, economics, and game theory.

9. Conclusion

This article has comprehensively explored 12 consensus algorithms, starting with an in-depth analysis of Proof of Work (PoW) and its limitations, such as low throughput, latency issues, high power consumption, and susceptibility to double spending attacks through potential forks. To address these limitations, various consensus protocols like Proof of Stake, Proof of Elapsed Work and Luck, and Delegated Proof of Stake were introduced to tackle the energy consumption issues inherent in PoW. The article delved into the competitive nature of PoW, leading to fork creation and prolonged wait times for transaction confirmation. The Proof of Elapsed Time, on the other hand, selects a node based on random waiting time, alleviating throughput and latency challenges. Bitcoin-NG, ByzCoin, and Algorand were discussed as solutions to enhance PoW's scalability, offering features such as consensus finality, instant transaction commitment, minimal computation, robust security, and perfect scalability. Furthermore, Raft and PBFT were explored, with Raft managing replicated logs and PBFT requiring two-thirds of all nodes to vote for the addition of the next block. Despite high throughput, PBFT faces scalability issues, a challenge addressed by Weighted Byzantine Fault Tolerance (WBFT). WBFT assigns initial weights to nodes, dynamically adjusting them based on behavior, resulting in improved performance by mitigating malicious activities. The success of Bitcoin in showcasing the potential of blockchain beyond financial transactions underscores the technology's versatility. Blockchain, as a distributed ledger replicated across nodes, poses unique challenges in achieving consensus. To address these challenges, the development of new consensus algorithms tailored to specific application requirements is essential. This may involve the creation of hybrid algorithms combining existing ones or the design of entirely new algorithms, potentially leveraging machine learning methodologies to meet the diverse needs of various applications.

Declaration of competing interest

Brij B Gupta is an associate editor for Cyber Security and Applications and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

CRediT authorship contribution statement

Ankit Kumar Jain: Writing – review & editing, Methodology. **Nishant Gupta:** Supervision, Writing – review & editing. **Brij B. Gupta:** Supervision, Writing – review & editing.

References

[1] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, Decentralized Bus. Rev. 21260 (2008).
[2] G. Karame, S. Capkun, Blockchain security and privacy, IEEE Secur. Privacy 16 (04) (2018) 11–12.
[3] Q. Zhou, H. Huang, Z. Zheng, J. Bian, Solutions to scalability of blockchain: a survey, IEEE Access 8 (2020) 16440–16455.
[4] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview, 2019, ArXiv preprint arXiv:1906.11078.
[5] S. Debnath, A. Chattopadhyay, S. Dutta, Brief review on journey of secured hash algorithms, in: 2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix), 2017, pp. 1–5.
[6] A. Maetouq, S.M. Daud, N.A. Ahmad, N. Maarop, N.N.A. Sjarif, H. Abas, Comparison of hash function algorithms against attacks: a review, Int. J. Adv. Comput. Sci. Appl. 9 (8) (2018).
[7] X. Zhang, R. Qin, Y. Yuan, F.Y. Wang, An analysis of blockchain-based bitcoin mining difficulty: techniques and principles, in: 2018 Chinese Automation Congress (CAC), 2018, pp. 1184–1189.
[8] D. MacKenzie, Pick a nonce and try a hash, London Rev. Books 41 (8) (2019) 35–38.
[9] J. Lu, J. Shen, P. Vijayakumar, B.B. Gupta, Blockchain-based secure data storage protocol for sensors in the industrial internet of things, IEEE Trans. Ind. Inf. 18 (8) (2021) 5422–5431.
[10] S. Nathan, C. Govindarajan, A. Saraf, M. Sethi, P. Jayachandran, Blockchain meets database: design and implementation of a blockchain relational database, 2019, ArXiv preprint arXiv:1903.01919.
[11] Y. Zhang, C. Xu, H. Li, H. Yang, X. Shen, Chronos: secure and accurate time-stamping scheme for digital files via blockchain, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6.
[12] G. Estevam, L.M. Palma, L.R. Silva, J.E. Martina, M. Vigil, Accurate and decentralized timestamping using smart contracts on the ethereum blockchain, Inf. Process. Manag.2021, 58, 3, 102471
[13] S. Dhumwad, M. Sukhadeve, C. Naik, K.N. Manjunath, S. Prabhu, A peer to peer money transfer using SHA256 and merkle tree, in: 2017 23rd Annual International Conference in Advanced Computing and Communications (ADCOM), 2017, pp. 40–43.
[14] C. Castellon, S. Roy, P. Kreidl, A. Dutta, L. Bölöni, Energy efficient merkle trees for blockchains, in: in 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2021, pp. 1093–1099.
[15] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, C. Qijun, A review on consensus algorithm of blockchain, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2017, pp. 2567–2572.
[16] S.J. Alsunaidi, F.A. Alhaidari, A survey of consensus algorithms for blockchain technology, in: 2019 International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1–6.
[17] V. Sharma, N. Lal, A novel comparison of consensus algorithms in blockchain, Adv. Appl. Math. Sci. 20 (1) (2020) 1–13.
[18] A. Begum, A. Tareq, M. Sultana, M. Sohel, T. Rahman, A. Sarwar, Blockchain attacks analysis and a model to solve double spending attack, Int. J. Mach. Learn. Comput. 10 (2) (2020) 352–357.
[19] G.N. Nguyen, N.H. Le Viet, M. Elhoseny, K. Shankar, B.B. Gupta, A.A. Abd El-Latif, Secure blockchain enabled cyber-physical systems in healthcare using deep belief network with resnet model, J. Parallel Distrib. Comput. 153 (2021) 150–160.
[20] S. Zhang, J.H. Lee, Double-spending with a sybil attack in the bitcoin decentralized network, IEEE Trans. Ind. Inf. 15 (10) (2019) 5715–5722.
[21] P. Swathi, C. Modi, D. Patel, Preventing sybil attack in blockchain using distributed behavior monitoring of miners, in: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1–6.
[22] K. Driscoll, B. Hall, M. Paulitsch, P. Zumsteg, H. Sivencrona, The real byzantine generals, in: The 23rd Digital Avionics Systems Conference (IEEE Cat. No. 04CH37576), 2, 2004, pp. 6–D.
[23] K. Driscoll, B. Hall, H. Sivencrona, P. Zumsteg, Byzantine fault tolerance, from theory to reality, in: Computer Safety, Reliability, and Security: 22nd International Conference, SAFECOMP 2003, Edinburgh, UK, September 23-26, 2003. Proceedings, 22, Springer Berlin Heidelberg, 2003, pp. 235–248.
[24] A. Al-Qerem, M. Alauthman, A. Almomani, B.B. Gupta, IoT transaction processing through cooperative concurrency control on fog-cloud computing environment, Soft Comput. 24 (2020) 5695–5711.
[25] P. Paul, P.S. Aithal, R. Saavedra, Blockchain technology and its types—a short review, 2021, Int. J. Appl. Sci. Eng. (IJASE), 9, 2, 189–200,

[26] M. Liu, K. Wu, J.J. Xu, How will blockchain technology impact auditing and accounting: permissionless versus permissioned blockchain, Curr. Issues Audit. 13 (2) (2019) A19–A29.

[27] M. Jovanovic, N. Kostić, I.M. Sebastian, T. Sedej, Managing a blockchain-based platform ecosystem for industry-wide adoption: the case of tradelens, Technol. Forecast. Soc. Change 184 (2022) 121981.

[28] M. Salimitari, M. Chatterjee, Y.P. Fallah, A survey on consensus methods in blockchain for resource-constrained IoT networks, Internet Things 11 (2020) 100212.

[29] Y. Xiao, N. Zhang, W. Lou, Y.T. Hou, A survey of distributed consensus protocols for blockchain networks, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1432–1465.

[30] S.M.H. Bamakan, A. Motavali, A.B. Bondarti, A survey of blockchain consensus algorithms performance evaluation criteria, Expert Syst. Appl. 154 (2020) 113385.

[31] X. Fu, H. Wang, P. Shi, A survey of blockchain consensus algorithms: mechanism, design and applications, Sci. China Inf. Sci. 64 (2021) 121101, doi:10.1007/s11432-019-2790-1.

[32] M.H. Nasir, J. Arshad, M.M. Khan, M. Fatima, K. Salah, R. Jayaraman, Scalable blockchains—a systematic review, Future Gener. Comput. Syst. 126 (2022) 136–162.

[33] B.B. Gupta, M. Quamara, An overview of internet of things (IoT): architectural aspects, challenges, and protocols, Concurrency Comput. Pract. Exper. 32 (21) (2020) e4946.

[34] J. Zhu, P. Liu, L. He, Mining information on bitcoin network data, in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017, pp. 999–1003.

[35] S. Park, S. Im, Y. Seol, J. Paek, Nodes in the bitcoin network: comparative measurement study and survey, IEEE Access 7 (2019) 57009–57022.

[36] H. Vranken, Sustainability of bitcoin and blockchains, Curr. Opin. Environ. Sustain. 28 (2017) 1–9.

[37] M.A. Fauzi, N. Paiman, Z. Othman, Bitcoin and cryptocurrency: challenges, opportunities and future works, J. Asian Finance Econ. Bus. 7 (8) (2020) 695–704.

[38] J. Taskinsoy, Bitcoin: a new digital gold standard in the 21st century?, 2021, Available at SSRN 3941857

[39] A.I. Sanka, R.C. Cheung, A systematic review of blockchain scalability: issues, solutions, analysis and future research, J. Netw. Comput. Appl. 195 (2021) 103232.

[40] T.M. Conaghy, R. Marques, A. Müller, D.D. Jonghe, T.M. Conaghy, G.M. Mullen, BigchainDB: a scalable blockchain database, White paper, BigChainDB (2016).

[41] A. Hafid, A.S. Hafid, M. Samih, Scaling blockchains: a comprehensive survey, IEEE Access 8 (2020) 125244–125262.

[42] Y. Zhang, D. Yang, G. Xue, CheaPay: an optimal algorithm for fee minimization in blockchain-based payment channel networks, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6.

[43] M. Zamani, M. Movahedi, M. Raykova, Rapidchain: scaling blockchain via full sharding, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 931–948.

[44] H. Dang, T.T.A. Dinh, D. Loghin, E.C. Chang, Q. Lin, B.C. Ooi, Towards scaling blockchain systems via sharding, in: Proceedings of the 2019 International Conference on Management of Data, 2019, pp. 123–140.

[45] M.S. Ferdous, M.J.M. Chowdhury, M.A. Hoque, A. Colman, Blockchain consensus algorithms: a survey, 2020, ArXiv preprint arXiv:2001.07091.

[46] N. Andola, S. Venkatesan, S. Verma, PoEWAL: a lightweight consensus mechanism for blockchain in IoT, Pervasive Mob. Comput. 69 (2020) 101291.

[47] H. Qin, Y. Cheng, X. Ma, F. Li, J. Abawajy, Weighted byzantine fault tolerance consensus algorithm for enhancing consortium blockchain efficiency and security, J. King Saud Univ.-Comput. Inf. Sci. 34 (10) (2022) 8370–8379.

[48] O.V. Portna, N. Iershova, D.A. Tereshchenko, O.R. Kryvitska, Economic business partnerships within Industry 4.0: new technologies in management.

[49] M. Sharma, S. Kamble, V. Mani, R. Sehrawat, A. Belhadi, V. Sharma, Industry 4.0 adoption for sustainability in multi-tier manufacturing supply chain in emerging economies, J. Clean. Prod. 281 (2021) 12501.

[50] H. Li, D. Han, M. Tang, Logisticschain: a blockchain-based secure storage scheme for logistics data, Mob. Inf. Syst. (2021) 2021.

[51] S. Biswas, K. Sharif, F. Li, Z. Latif, S.S. Kanhere, S.P. Mohanty, Interoperability and synchronization management of blockchain-based decentralized e-health systems, IEEE Trans. Eng. Manag. 67 (4) (2020) 1363–1376.

[52] S. Khanam, S. Tanweer, S.S. Khalid, Future of internet of things: Enhancing cloud-based IoT using artificial intelligence, Int. J. Cloud Appl. Comput. (IJCAC) 12 (1) (2022) 1–23.

[53] S. Aggarwal, N. Kumar, M. Alhussein, G. Muhammad, Blockchain-based UAV path planning for healthcare 4.0: current challenges and the way ahead, IEEE Netw. 35 (1) (2021) 20–29.

[54] A.R. Rajput, Q. Li, M.T. Ahvanooey, A blockchain-based secret-data sharing framework for personal health records in emergency condition, in: Healthcare, 9, 2021, p. 206.

[55] B. Bera, A.K. Das, A.K. Sutrala, Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in internet of drones environment, Comput. Commun. 166 (2021) 91–109.

[56] C.L. Chen, Y.Y. Deng, W. Weng, M. Zhou, H. Sun, A blockchain-based intelligent anti-switch package in tracing logistics system, J. Supercomput. 77 (2021) 7791–7832.

[57] R. Gupta, A. Kumari, S. Tanwar, Fusion of blockchain and artificial intelligence for secure drone networking underlying 5G communications, Trans. Emerg. Telecommun. Technol. 32 (1) (2021) e4176.

[58] Z. Shahbazi, Y.C. Byun, A procedure for tracing supply chains for perishable food based on blockchain, machine learning and fuzzy logic, Electronics 10 (1) (2020) 41.

[59] M.A. Kiran, S.K. Pasupuleti, R. Eswari, Efficient pairing-free identity-based signcryption scheme for cloud-assisted IoT, Int. J. Cloud Appl. Comput. (IJCAC) 12 (1) (2022) 1–15.

[60] W. Ren, X. Wan, P. Gan, A double-blockchain solution for agricultural sampled data security in internet of things network, Future Gener. Comput. Syst. 117 (2021) 453–461.

[61] S. Saurabh, K. Dey, Blockchain technology adoption, architecture, and sustainable agri-food supply chains, J. Clean. Prod. 284 (2021) 124731.

[62] A. Hariharasudan, I. Otola, Y. Bilan, Reactive power optimization and price management in microgrid enabled with blockchain, Energies 13 (23) (2020) 6179.

[63] H.T. Doan, J. Cho, D. Kim, Peer-to-peer energy trading in smart grid through blockchain: a double auction-based game theoretic approach, IEEE Access 9 (2021) 49206–49218.

[64] M.B. Mollah, J. Zhao, D. Niyato, K.Y. Lam, X. Zhang, A. M. Ghias, L. Yang, Blockchain for future smart grid: a comprehensive survey, IEEE Internet Things J. 8 (1) (2020) 18–43.

[65] F.B. Schneider, Implementing fault-tolerant services using the state machine approach: a tutorial, ACM Comput. Surv. (CSUR) 22 (4) (1990) 299–319.

[66] M. Burrows, The chubby lock service for loosely-coupled distributed systems, in: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, 2006, pp. 335–350.

[67] R. Kumar, S.K. Singh, D.K. Lobiyal, K.T. Chui, D. Santaniello, M.K. Rafsanjani, A novel decentralized group key management scheme for cloud-based vehicular IoT networks, Int. J. Cloud Appl. Comput. (IJCAC) 12 (1) (2022) 1–34.

[68] C. Dwork, U. Feige, J. Kilian, M. Naor, S. Safra, Low communication perfect zero knowledge two provers proof systems, in: Crypto92, volume 740, Springer Verlag, p. 1992. Lecture Notes in Computer Science. 215-227

[69] A. Back, Hashcash-a denial of service counter-measure, 2002

[70] M.S. Ferdous, M.J.M. Chowdhury, M.A. Hoque, A survey of consensus algorithms in public blockchain systems for crypto-currencies, J. Netw. Comput. Appl. 182 (2021) 103035.

[71] Y. Sompolinsky, A. Zohar, Accelerating Bitcoin's Transaction Processing. Fast Money Grows on Trees, Not Chains, Cryptology ePrint Archive, 2013.

[72] A. Gervais, G.O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, S. Capkun, On the security and performance of proof of work blockchains, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 3–16.

[73] A. Kiayias, G. Panagiotakos, Speed-Security Tradeoffs in Blockchain Protocols, Cryptology ePrint Archive, 2015.

[74] J. Garay, A. Kiayias, N. Leonardos, The bitcoin backbone protocol: analysis and applications, in: Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Springer Berlin Heidelberg, 2015, pp. 281–310. Proceedings, Part II

[75] K.J. O'Dwyer, D. Malone, Bitcoin mining and its energy footprint, 2014,

[76] J. Debus, Consensus methods in blockchain systems, Frankfurt School of Finance & Management, Blockchain Center, Tech. Rep., 2017.

[77] Z. Zheng, S. Xie, H.N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: a survey, Int. J. Web Grid Serv. 14 (4) (2018) 352–375.

[78] D. Larimer, Delegated Proof-of-Stake (DPOS), Bitshare whitepaper, 2014.

[79] J. Fu, W. Zhou, S. Zhang, Fabric blockchain design based on improved SM2 algorithm, Int. J. Semantic Web Inf. Syst. (IJSWIS) 19 (1) (2023) 1–13.

[80] P4Titan, "Slimcoin: A Peer-To-Peer Crypto-Currency with Proof-of-Burn,". 2014,

[81] Hyperledger, https://www.hyperledger.org.Accessed: 2022-12-07.

[82] M. Salimitari, M. Chatterjee, A survey on consensus protocols in blockchain for IoT networks, 2018, ArXiv preprint arXiv:1809.05613.

[83] S. Micali, M. Rabin, S. Vadhan, Verifiable random functions, in: 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039), 1999, pp. 120–130.

[84] I. Eyal, A.E. Gencer, E.G. Sirer, R.V. Renesse, Bitcoin-NG: a scalable blockchain protocol, in: 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), 2016, pp. 45–59.

[85] E.K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, B. Ford, Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing, USENIX Association, 2016.

[86] E. Syta, I. Tamas, D. Visher, D.I. Wolinsky, P. Jovanovic,. L. Gasser, B. Ford, Keeping authorities' honest or bust' with decentralized witness cosigning, in: 2016 IEEE Symposium on Security and Privacy (SP), May 2016, pp. 526–545.

[87] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm (extended version), in: Proceeding of USENIX Annual Technical Conference, USENIX ATC, 2014, pp. 19–20.

[88] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, in: Concurrency: The Works of Leslie Lamport, 2019, pp. 203–226.

[89] A. Tiwari, R. Garg, Adaptive ontology-based IoT resource provisioning in computing systems, Int. J. Semantic Web Inf. Syst. (IJSWIS) 18 (1) (2022) 1–18.

[90] L. Lamport, Paxos made simple, ACM SIGACT News (Distributed Computing Column) 32 (4) (2001) 51–58.

[91] M. Castro, B. Liskov, Practical byzantine fault tolerance, OsDI 99 (1999) (1999) 173–186.

[92] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, Algorand: scaling byzantine agreements for cryptocurrencies, in: Proceedings of the 26th Symposium on Operating Systems Principles, 2017, pp. 51–68.

[93] M. Vukolić, The quest for scalable blockchain fabric: proof-of-work vs. BFT replication, in: International Workshop on Open Problems in Network Security, Springer, Cham, 2015, pp. 112–125.

[94] S. Bano, M. AlBassam, G. Danezis, The road to scalable blockchain designs, USENIX; login: magazine 42 (4) (2017) 31–36.

[95] K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, R. Wattenhofer, On scaling decentralized blockchains, in: International Conference on Financial Cryptography and Data Security, Springer, Berlin, Heidelberg, 2016, pp. 106–125.
[96] S. Zhang, J.H. Lee, Double-spending with a sybil attack in the bitcoin decentralized network, IEEE Trans. Ind. Inf. 15 (10) (2019) 5715–5722.
[97] M.G. Raj, S.K. Pani, Chaotic whale crow optimization algorithm for secure routing in the IoT environment, Int. J. Semantic Web Inf. Syst. (IJSWIS) 18 (1) (2022) 1–25.
[98] S. Sayeed, H. Marco-Gisbert, Assessing blockchain consensus and security mechanisms against the 51% attack, Appl. Sci. 9 (9) (2019).