Cybersecurity

REVIEW

Open Access

# Evolution of blockchain consensus algorithms: a review on the latest milestones of blockchain consensus algorithms

Ziad Hussein[1*], May A. Salama[1] and Sahar A. El-Rahman[1]

**Abstract**

Blockchain technology has gained widespread adoption in recent years due to its ability to enable secure and transparent record-keeping and data transfer. A critical aspect of blockchain technology is the use of consensus algorithms, which allow distributed nodes in the network to agree on the state of the blockchain. In this review paper, we examine various consensus algorithms that are used in blockchain systems, including proof-of-work, proof-of-stake, and hybrid approaches. We go over the trade-offs and factors to think about when choosing a consensus algorithm, such as energy efficiency, decentralization, and security. We also look at the strengths and weaknesses of each algorithm as well as their potential impact on the scalability and adoption of blockchain technology.

**Keywords** Blockchain, Consensus, Proof of work, Proof of stake, Decentralization

## Introduction

Blockchain is one of the most promising emerging technologies in the 21st century. It offers significant benefits, such as decentralization, non-tampering, non-forgery, and traceability, making it ideal for storing and securing important anti-counterfeiting data (Guo and Yu 2022). It also has the potential to solve security issues related to data tampering and loss in traditional centralized endorsement agencies, as well as improve the efficiency of transaction processing in various fields, such as finance, medical, Internet of Things, property rights protection, privacy protection, etc. Blockchain technology has been gaining attention for its ability to create great value. It originated from the 2008 paper "Bitcoin: A Peer-to-Peer Electronic Cash System" written by "Satoshi Nakamoto" (Nakamoto 2008). The creation of the genesis block in Bitcoin in 2009 marked the official birth of blockchain technology, signaling the emergence of a new scientific field and an innovative distributed technology. The consensus algorithm is a protocol or mechanism that is used to achieve agreement among the nodes in a distributed network (Xiao et al.2019). In a blockchain network, the nodes are computers or devices that store and maintain a copy of the blockchain, and the consensus algorithm is used to ensure that all the nodes have the same view of the blockchain and agree on the order of transactions. This is important because it allows the network to maintain a single, consistent, and tamper-evident ledger of transactions without the need for central authority. The drawbacks of blockchain technology have started to materialize. For example, the ability to manage redundant transactions is incompetent in terms of performance and scaling of network (Ammous and Saifedean 2016). Moreover, sometimes the data throughput is too small and the capacity of defending the blockchain from malicious nodes is limited. One of the most substantial problems of blockchain technology is the power consumption; there's an urge need for the intensive consumption of the computational power (Denisova 2019). This problem is considered a stumbling stone in the

*Correspondence:
Ziad Hussein
z.hussien56795@feng.bu.edu.eg
[1] Electrical Engineering Department, Faculty of Engineering-Shoubra, Benha University, Cairo, Egypt

evolution path of this field. Solving this problem is considered one of the most important research points. Consensus algorithms are not a new topic, in fact, it predates the era of blockchain itself. One of the leading influences in this field can be tracked back to the late fifties of the last century, when it had been proposed one of the consensus algorithms when it had been used as a part of probability function (Xiong et al. 2022). Consensus algorithms are classified under two main categories; one of them considers the existence of malicious nodes and one does not. The one it does is called nodes and one does not. The one that does is called Byzantine fault-tolerant and the one that does not is called non-Byzantine fault-tolerant. One of the most well-known non-Byzantine was introduced in (Oki and Liskov 1988), while in 1989 Lamport proposed the Paxos algorithm. In 2008, (Nakamoto 2008) introduced Bitcoin using the Proof-of-Work (PoW) algorithm which, on the contrary, considered the malicious nodes. There are various types of consensus algorithms, and each one has its own strengths and limitations. Some of the most widespread consensus algorithms used in blockchain technology are Proof-of-Work (PoW) (Gervais et al. 2016). This is the earliest consensus algorithm used by Bitcoin, and it relies on miners (nodes in the network) to compete to solve cryptographic problems to validate transactions and create new blocks in the blockchain. PoW is secure and decentralized, but it requires a lot of energy and can be sluggish and incompetent. Proof-of-Stake (PoS) is a fresher consensus algorithm that allows nodes in the network to authenticate transactions and create new blocks based on their stake (Bentov et al. 2014), or the amount of cryptocurrency they hold. PoS is more energy-efficient and scalable than PoW, but it can be more susceptible to centralization and attacks by wealthy nodes. Delegated Proof-of-Stake (DPoS) is a variant of PoS where the nodes in the network vote to elect a small number of "delegates" who are responsible for validating transactions and creating new blocks (Xu et al. 2019a). DPoS is faster and more scalable than PoW or PoS, but it can be less decentralized and more vulnerable to corruption or collusion.

Overall, consensus algorithms are a vital part of blockchain technology, and they play a major role in guaranteeing the security, decentralization, and scalability of blockchain networks. Different consensus algorithms have several trade-offs, and the choice of algorithm can have significant outcomes for the assets and performance of a blockchain network. In this section we have introduced the topic, while in section 2 we are going to present an overview on the blockchain. In section 3, we are going to explain some of the most substantial consensus algorithms, and various comparisons will be conducted among different types of consensus algorithms with respect to different criteria. In section 4, we will demonstrate some future improvements. Finally, section 5 concludes our research and focuses on the difference between our paper and others' researches.

## Blockchain overview

In this section we are going to discuss blockchain architecture, how it works and illustrate the core of the blockchain technology.

### Architecture

The blockchain is a type of Distributed Ledger Technology (DLT) (El loini and Bahl 2018) that allows the secure, transparent, and unalterable storage of data. It consists of a network of computers, called nodes, that preserve a shared, aligned record of trades. These transactions are pushed into blocks, which are linked together in a linear chain, with each block containing a timestamp and a link to the previous block. This arrangement allows for the creation of a secure, decentralized database that is repellent to tampering and revision. The blockchain architecture is aimed to be decentralized implying that it is not regulated by a single central authority. Instead, the network is preserved by a network of nodes that work together to authenticate and record transactions. This decentralized structure permits the transfer of digital assets, such as cryptocurrency, without the presence of intermediaries, such as banks or other financial organizations (El loini and Bahl 2018; Wright and De Filippi 2015). The blockchain is also created to be secure. Each block in the chain is secured using cryptographic techniques, making it too complicated for anyone to modify the data once it has been recorded. Additionally, the decentralized nature of the network means that no single node can gain control of the network and manipulate the data (Zhang et al. 2019). The transparency of the blockchain is another key characteristic of its architecture. Because the network is decentralized and accessible, anyone can see the transactions that have been logged on the blockchain. This allows for better clarity and liability, as users can see exactly where their assets are and how they are being used (Sunny et al. 2020). The block header is a vital part of the blockchain architecture (Puthal et al. 2018). It is the first item that is accumulated in a block and contains several important pieces of information, including the following:

- Reference link to the prior block, also known as the "parent" block. This link is what creates the chain of blocks. The blocks are connected through hash codes (Fu et al. 2021).
- Timestamp, which specifies when the block was built.

- Proof of work, which is a mathematical problem that must be cracked in order to insert a new block to the chain. This proof of work is what permits the network to reach consensus on the state of the blockchain and inhibits the chain from being altered.
- Merkle root, which is a hash of all the operations in the block (Merkle 1988). This allows users to validate the truthfulness of the transactions without having to transfer the entire block.
- Nonce, (number used once) which is an arbitrary number that is used in the PoW calculation, and the miners are trying to find it. It is a 32-bit number that usually takes 10 min to be guessed (Baldominos and Saez 2019).

A Merkle tree, also known as a binary hash tree, is a data structure that is used in the blockchain to effectively verify the integrity of large sets of data. It is named after Ralph Merkle, who created the idea in the 1980s (Merkle 1988). The formation of a Merkle tree is defined by the way that the hashes of the data are arranged and mixed. In a Merkle tree, the separate pieces of data are hashed and positioned in a binary tree structure, with each leaf node comprising the hash of a single piece of data. These leaf nodes are then combined in pairs, with each parent node containing the hash of its two child nodes. This process is repeated until there is only one final "root" node, which contains the combined hash of all the data in the tree. Figure 1 depicts the Merkle tree. The most important benefit of using a Merkle tree is that it permits effective authentication of the reliability of large sets of data. In the blockchain, a Merkle tree can be used to certify that an operation has been incorporated in a block without having to transfer the entire block. Since the root node of the tree contains the combined hash of all the operations. Therefore, a user only requires transferring the root node to verify the integrity of the data (Liu et al. 2021).

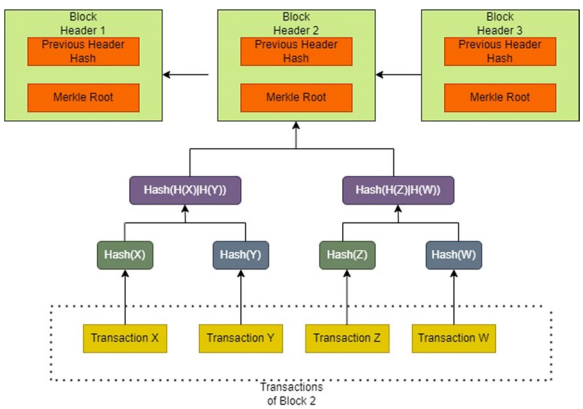


**Fig. 1** Merkle tree architecture

The block body is the part of a block that stores the actual data of the transactions that are being recorded on the blockchain. The block body typically includes the following information (Ismail and Materwala 2019):

- A list of the transactions that are being recorded in the block. This will typically include information such as the sender and recipient of the transaction, the amount of the transaction, and any other relevant data.
- The cryptographic signatures of the transactions, which are used to verify the authenticity of the transactions and ensure that they have not been tampered with.
- Any other relevant data, such as transaction fees or other metadata.

In the block body of the Merkle tree, all information of the transactions is being processed. Every leaf node of the tree stores the transaction information, and it is paired by a hash calculation and combined to generate the hash until obtaining the root node of the tree. The hash value of the tree is able to detect any tampering because if any leaf has been tampered with, that will definitely change the hash value of the tree root (Ismail and Materwala 2019). The Merkle tree structure is the failsafe of the blockchain because it ensures the security of the information in the blockchain. In the blockchain, the word "node" is broadly being used. It is simply a machine that performs computations. The node in the blockchain is behaving in P2P style (Li et al. 2018). The network is observing and coordinating the operations of the nodes in a decentralized conduct. The key task of the node in the blockchain is to check the information validity and store the correct data. Nodes can be classified into three types: the mining node, which is accountable for creating and issuing the new blocks; the broadcasting nodes, which is sending the information of transactions and receiving limited amount of data; and the complete node, which is responsible for issuing transactions, verifying the data, and propagating transactions (Perard et al. 2018).

### Working theory of blockchain

The blockchain (BC) is a decentralized, distributed database that is used to preserve a constantly expanding list of records, called blocks. Each block contains a timestamp and a link to the prior block, making it challenging to modify the data once it has been Preserved. This structure permits the creation of a secure, visible, and unchanging ledger of trades that is handled by a network

of workstations on the internet, rather than a single main authority (Singhal et al. 2018). It is valuable to know the fundamental concepts of decentralization consensus and Cryptography. Decentralization means the blockchain is not controlled by one main authority. Instead, it is retained by a network of computers, known as nodes, that work simultaneously to prove and record transactions. This decentralized structure allows the efficient and secure transfer of digital assets without the need for mediators. The usage of mathematical algorithms to encrypt and secure data is called cryptography. In BC, cryptographic methods are used to safeguard each block in the chain and verify the authenticity of transactions (Singhal et al. 2018). This makes it difficult for anyone to alter the data once it has been recorded on the blockchain while consensus is in the process of attaining an agreement on the status of the blockchain (Xiong et al. 2022). This is done by using (PoW), which necessitates the nodes to compete to solve a mathematical problem to attach a new block to the chain. The first node to solve the problem is permitted to add the new block, and the other nodes then validate that the block is acceptable before adding it to their own copies of the blockchain. Eventually BC is a decentralized secure and transparent system for saving and validating transactions. It uses cryptographic methods and consensus-based tactics to maintain a secure and unchanging ledger. The basic mechanics of how the blockchain acts can be condensed as follows (Singhal et al. 2018):

- A transaction is started by a user and disseminated to the network of nodes.
- Public vs. private: Access to a public blockchain is open to everyone, but a private blockchain is only available to a select number of individuals.
- Permissioned vs. permissionless: A permissioned blockchain needs users to be authorized in order to engage in the network, whereas a permissionless blockchain allows anybody to join in the network and validate transactions.
- Decentralized vs. centralized: A decentralized blockchain is one that is dispersed among a network of nodes as opposed to a centralized blockchain, which is one that is managed by a single entity.
- Federated vs. consortium: A federated blockchain is one that is managed by a number of entities, but a consortium blockchain is managed by a number of carefully chosen entities.

### Classes and structure of blockchain

There are several concepts and classifications for the construction of the blockchain. These classifications and models are based on variables such the network type, the consensus process employed, and the degree of decentralization. The following are some of the most well-known blockchain structure types and classifications (Guo and Yu 2022):

- Public vs. private: Access to a public blockchain is open to everyone, but a private blockchain is only available to a select number of individuals.
- Permissioned vs. permissionless: A permissioned blockchain needs users to be authorized in order to engage in the network, whereas a permissionless blockchain allows anybody to join in the network and validate transactions.
- Decentralized vs. centralized: A decentralized blockchain is one that is dispersed among a network of nodes as opposed to a centralized blockchain, which is one that is managed by a single entity.
- Federated vs. consortium: A federated blockchain is one that is managed by a number of entities, but a consortium blockchain is managed by a number of carefully chosen entities.

Blockchain can also be described as a layered system. The majority of Blockchains can be designed with the following layers: network layer, data layer, consensus layer, incentive layer, contract layer, and application layer as shown in Fig. 2.

The data layer, network layer, consensus algorithm, incentive layer, contract layer, and application layer are some of the layers that to make up the blockchain.

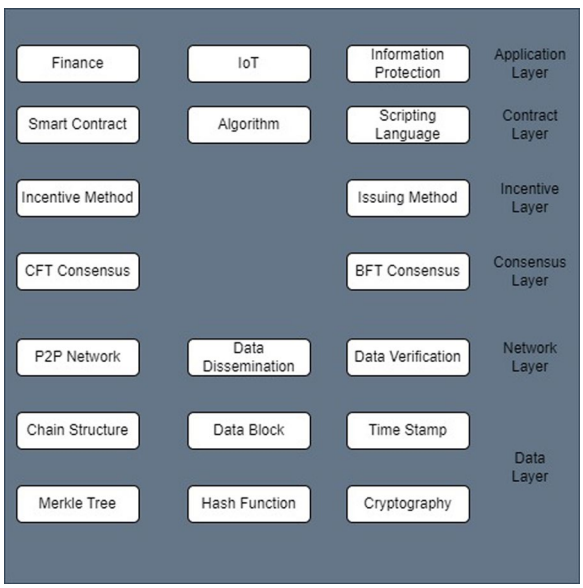


**Fig. 2** Layered system of blockchain

Data blocks, a linked list, Merkle trees, and other data structures that make use of timestamps, hash functions, and cryptography make up the data layer. It provides the framework for blockchain management, organization, and data storage. All nodes in the chain are connected via the network layer using a peer-to-peer network mechanism, allowing them to transact, send, and verify data. The blockchain's basic technology, the consensus algorithm, determines which nodes have the right to record transactions and enables them to swiftly agree on the information included in a block. This ensures the consistency and security of the data while also improving the blockchain's computational efficiency. By incorporating rewards and punishments into the blockchain's distribution mechanism, the incentive layer incentivizes nodes to provide services. The contract layer consists of smart contracts and algorithms that are executed automatically when certain conditions are met, allowing for the customization of blockchain transactions. The application layer combines the underlying structure, script code, and smart contracts to allow blockchain to be used in a variety of real-world scenarios. Based on data access authority, blockchain can be classified into three types (Sheth and Dattani 2019): public, consortium, and private. Public chains allow nodes to join and leave the network without requiring permission, but they are slow in terms of transaction processing speeds and low overall performance. Examples of public chain applications include Bitcoin and Ethereum (Yang et al. 2020). Consortium chains require nodes to register and be approved by a central organization before joining the network. They have lower degrees of visibility than public chains and are suitable for enterprises or companies that form consortia. Nodes in consortium chains do not completely trust each other and require consensus algorithms to reach agreement. Private chains are controlled by a single internal entity and only allow access to selected and verified participants. They have the lowest degree of decentralization but the fastest transaction processing speeds. Table 1 compares the characteristics of the three types of blockchain. In brief, this section reviewed various blockchain concepts and classifications based on variables such as network type, consensus process, and degree of decentralization. Public vs. private blockchain structures, permissioned vs. permissionless, decentralized vs. centralized, and federated vs. consortium were also covered. Blockchain's layers, which include the data layer, network layer, consensus layer, incentive layer, contract layer, and application layer were explained as well. Furthermore, blockchain types based on data access authority were considered in the review.

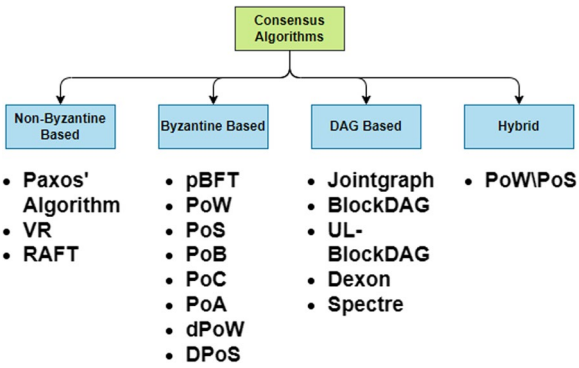


**Fig. 3** Categories of consensus algorithms

## Consensus algorithms

Consensus is the process of reaching agreement among a group of people or entities on a specific decision or action. In a blockchain, consensus is used to guarantee that all nodes on the network agree on the current state of the network and the authenticity of transactions (Xiong et al. 2022). This is vital for preserving the security and integrity of the blockchain. Different blockchain platforms use different algorithms, such as proof of work, proof of stake, or proof of authority to achieve consensus among the nodes on the network.

### Concept of consensus (historical background)

Under certain conditions, Edmund Eisenberg and David Gale conducted a study in 1959 on how individuals with subjective consciousness in the same space can reach a consistent consensus probability distribution. This is known as the consistency problem, and it is also referred to as the consensus problem. The study at the time only focused on scenarios where the number of nodes was limited and trustworthy, but this had limitations and was not suitable for open internet scenarios. Satoshi Nakamoto later extended the consistency problem to the internet environment with open scenes and massive nodes in the Bitcoin system, proposing the Byzantine Generals problem, which is known as the Byzantine Generals problem. This issue, known as Byzantine failures, is significant in the field of blockchain technology. The consensus algorithm ensures data consistency among nodes for a specific proposal. Different consensus algorithms have varying capabilities for ensuring that nodes receive balanced accounting rights. An excellent consensus algorithm can keep the blockchain network active and provide a steady stream of effective computing power to the entire network, whereas a poorly designed algorithm can cause the entire network to become easily paralyzed when attacked. Consensus algorithms are classified into non-Byzantine fault-tolerant algorithms, Byzantine

fault-tolerant algorithms, DAG based, and Hybrid. Figure 3 shows the different categories of consensus algorithms, in addition to the well-known algorithms under each of these categories.

### Non-Byzantine consensus algorithms

Non-Byzantine error is a type of system failure that occurs in distributed systems without the presence of malicious nodes. This can include issues such as machine downtime and node reporting errors (Xiong et al. 2022;Han and Gao 2020). Non-Byzantine fault-tolerant algorithms are designed to handle these types of errors, but they cannot guarantee the security of data and system stability when malicious nodes are present. Therefore, non-Byzantine fault-tolerant algorithms are typically only used in closed environments with high credibility between nodes, such as consortium chains or private chains. These algorithms offer high performance and strong tolerance for non-Byzantine errors.

#### *Paxos' algorithm*

The Paxos algorithm is a consensus algorithm that enables distributed network nodes to reach agreement on a proposed value (Lamport 2001). It was first introduced by computer scientist Leslie Lamport in 1998 and it has been used in many distributed systems. Unlike other consensus algorithms Paxos does not depend on a central authority to organize the consensus process. Instead, it uses messages exchanged between nodes to reach consensus on a proposed value. Paxos is fault-tolerant, which means that it can remain functional even when some nodes in the network fail or behave maliciously (De Prisco et al. 2000). The algorithm uses three types of nodes: proposers, which propose values for data items; acceptors, which evaluate and accept or reject proposed values; and learners, which receive accepted values and update their own local copies of the data. There are many variations and optimizations of the Paxos algorithm, but it remains a powerful tool for achieving consensus in distributed systems. The Paxos algorithm is a complex algorithm, but its basic workflow can be broken down into the following steps (Lamport 2001):

- A proposer node suggests a value for a specific data item. This value is broadcast to a set of acceptor nodes.
- The acceptor nodes receive and evaluate the proposed value. If they agree that the proposed value is correct, they send a message to the proposer indicating that the value has been accepted.
- If the proposed value is accepted by the majority of acceptor nodes, the proposer sends a message to a group of learner nodes indicating that the proposed value has been accepted by the majority of acceptors.
- When the accepted value is received, the learner nodes update their local copies of the data item with the accepted value.
- If a learner node receives a different value from a master node for the same data item, it must repeat the process from step 2 to reach consensus on the correct value.

Paxos' algorithm uses a series of messages exchanged between nodes to reach consensus on a proposed value (Lamport Lamport 2001; De Prisco et al. 2000). The algorithm is designed to be fault-tolerant, meaning that it can continue to function even if some nodes in the network fail or behave maliciously. This allows the algorithm to ensure that all nodes in the network agree on the same value for a given data item, Fig. 4 shows the algorithm flow.

#### *VR consensus algorithm*

Viewstamped Replication (VR) is a distributed consensus algorithm that allows nodes in a network to agree on the order and integrity of transactions in a distributed database. Each node maintains a log of transactions and a current "view" number, which indicates its current state (Liskov and Cowling 2012). When a node wants to propose a new transaction, it sends a request to a designated primary node, which is responsible for ordering the transactions and broadcasting them to the other nodes. Figure 5 illustrates the VR algorithm. If the primary node fails or becomes unavailable, the other nodes can initiate a change of view process to select a new primary node that will continue the process of ordering and broadcasting transactions, using the log and view numbers from

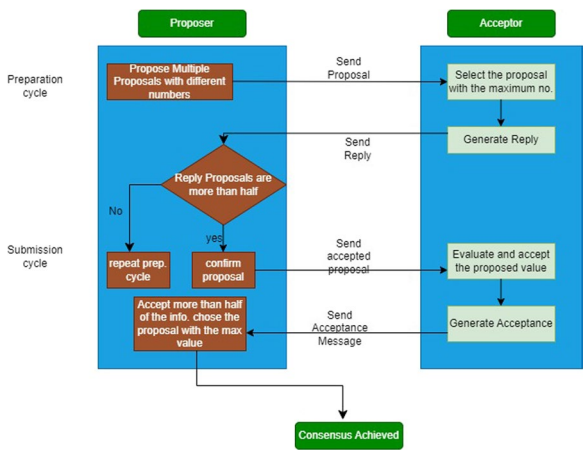


**Fig. 4** Flow chart of Paxos algorithm

**Table 1** Characteristics of blockchain types

| Feature | Types | | |
|---|---|---|---|
| | **Public** | **Consortium** | **Private** |
| Pros | Anyone can join and participate without permission or identity verification (Open) | Transactions are processed faster than public blockchains due to lower network latency and higher throughput (Faster) | Transactions are faster, scalable, efficient, secure and fully controlled by a single node |
| | All transactions are publicly visible and verifiable by anyone (Transparent) | Transactions can handle higher volumes than public blockchains due to lower computational complexity and resource consumption (Scalable) | Lower network latency and higher throughput make transactions faster than public blockchains |
| | Transactions are validated by a large number of nodes using a cryptographic consensus mechanism that prevents double-spending, fraud, or censorship (Secure) | Transactions consume less energy than public blockchains due to simpler consensus mechanisms that do not require intensive proof-of-work or proof-of-stake algorithms (Efficient) | Lower computational complexity and resource consumption make transactions scalable to higher volumes than public blockchains |
| | Transactions are permanently recorded on a distributed ledger that cannot be altered or erased by anyone (Immutable) | Transactions are validated by a smaller number of nodes using a cryptographic consensus mechanism that prevents double-spending, fraud, or censorship (Secure) | Simpler consensus mechanisms that do not require intensive algorithms make transactions efficient and consume less energy than public blockchains |
| | Transactions are executed according to predefined rules that cannot be changed by anyone without consensus from all nodes (Censorship-resistant) | Transactions are executed according to predefined rules that can be changed by agreement from a subset of nodes (More control over governance and consensus) | Cryptographic consensus mechanism that prevents double-spending, fraud or censorship makes transactions secure |
| | | | Predefined rules that can be changed by decision from a single node give full control over governance and consensus |
| Cons | High electricity consumption and processing power required for consensus | Less transparent and democratic than public blockchain | Highly centralized and dependent on a single authority |
| | Low scalability due to limited throughput and high latency | Prone to collusion or corruption among the controlling entities | Lacks immutability and security compared to public blockchain |
| | Vulnerable to 51% attacks if a single entity gains majority control over the network | May face legal or regulatory challenges due to cross-border transactions | May not benefit from network effects or innovation due to limited participation |
| Usage Scenario | Cryptocurrencies: Online payments without intermediaries or trusted third parties (e.g., Bitcoin) | Cross-border payments (e.g., Ripple): A global payment network that enables fast, cheap, and secure transactions across different currencies and jurisdictions | Enterprise solutions: IBM Blockchain Platform - A platform for developing, deploying, and managing private blockchain networks for business use cases such as trade finance, asset tracking, digital identity, etc |
| | Smart contracts: Decentralized applications that run on self-executing contracts (e.g., Ethereum) | Supply chain management (e.g., Hyperledger Fabric): A framework for building enterprise-grade blockchain solutions that enable transparency, traceability, and efficiency across complex supply chains | Banking and finance: Corda - A distributed ledger platform designed for financial institutions that enables secure transactions with smart contracts |
| | Decentralized applications: Applications that run on distributed networks without centralized servers or authorities (e.g., Uniswap) | Identity verification (e.g., Sovrin): A decentralized identity network that enables self-sovereign identity management using verifiable credentials | Healthcare records: MedRec - A prototype for managing electronic medical records using blockchain technology |
| Centralization Level | Network and data are not controlled by any single entity | Anyone can join and participate without permission or identity verification | Network and data are controlled by a group of entities |
| | All nodes have equal rights and responsibilities | Some nodes have more rights and responsibilities than others | Only one node has all rights and responsibilities |
| Participants | Any node can join and do transactions without permission or identity verification | Members only can join and do transactions with permission or identity verification by a group of organizations that jointly manage the network | Individual or company can join and do transactions with permission or identity verification by a single entity that owns |

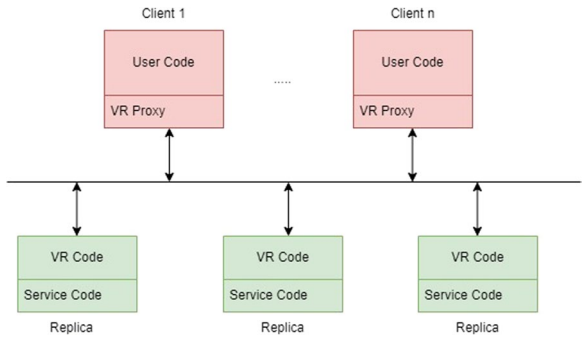


**Fig. 5** VR architecture

the previous primary node to ensure consistency. VR is able to handle node failures and network partitions without sacrificing consistency or security, making it suitable for use in distributed systems where strong consistency and fault tolerance are required, such as blockchain networks. However, it may be slower and less scalable than other algorithms like PoW and PoS. VR was first developed in 1988 by Liskov and has since been used in a variety of applications, including distributed databases and file systems, as well as blockchain networks (Oki and Liskov 1988).

#### *RAFT algorithm*

RAFT is a distributed consensus algorithm that allows a group of nodes (computers) in a distributed system to reach agreement on a single value or state of the system. It was developed as an alternative to other consensus algorithms, such as Paxos, and is designed to be easier to understand and implement. In RAFT, the nodes in the system are divided into two types: leaders and followers (Kim et al. 2021). The leader is responsible for proposing new values or states for the system and for replicating those values to the followers. The followers are responsible for accepting or rejecting the proposals based on a set of rules, and for voting for a new leader if the current leader becomes unavailable. Figure 6 shows the general architecture of the algorithm (Tian et al. 2021). The process of reaching consensus in RAFT works as follows: the leader proposes a new value or state for the system and broadcasts it to the followers. The followers accept or reject the proposal based on a set of rules. If a majority of followers accept the proposal, it is considered to be "committed". The leader then replicates the committed value to the followers and updates its own log of committed values. If the leader becomes unavailable, the followers can initiate a leader election process to select a new leader as shown in Fig. 7. The new leader is chosen based on the log of committed values, with the node that has the most up-to-date log being selected as the new leader (Kim et al. 2021). One key feature of RAFT is that it provides strong consistency (Hu and Liu 2020), meaning that all nodes in the system will eventually agree on the same value or state. It also provides fault tolerance, as the system can continue to operate even if some nodes fail or become unavailable. RAFT is a popular consensus algorithm that is widely used in distributed systems such as distributed databases and distributed file systems. It is well-known for its simplicity and ease of implementation, making it an excellent choice for systems requiring solid consistency and fault tolerance. RAFT has specific applications such as distributed key-value stores, distributed configuration management systems, and distributed file systems (Le Brun et al. 2021). It is also used as a foundation for other distributed systems such as distributed databases and distributed messaging systems. To summarize, this section discussed three non-Byzantine consensus algorithms, namely Paxos, VR, and RAFT. Non-Byzantine errors are failures that occur in distributed systems without the presence of malicious nodes.

These algorithms are designed to handle these types of errors but cannot guarantee the security of data and system stability when malicious nodes are present. Paxos and VR are fault-tolerant algorithms that are suitable for use in distributed systems where strong consistency and fault tolerance are required, such as blockchain networks. Paxos is a complex and inefficient algorithm that relies too much on messages exchanged between nodes to reach consensus on a proposed value. We recommend VR, which allows nodes to agree on the order and integrity of transactions in a distributed database without too much overhead. RAFT is another consensus algorithm that is designed to be easier to understand and implement than Paxos, but it has some drawbacks. It gives too much power to the leader, who is responsible for proposing new values or states for the system, and the followers have to accept or reject the proposals based on a set of rules. If the leader becomes unavailable, the followers have to initiate a leader election process to select a new leader, which can cause delays and conflicts.

### Byzantine-based consensus algorithms

Byzantine fault-tolerant algorithms (BFTs) are considered the ability of a distributed network to achieve consensus on a value, even when some nodes in the network do not respond correctly or at all.

The purpose of a BFT mechanism is to protect against system failures by using a collective decision-making process that considers the input of both correct and faulty nodes, in order to minimize the impact of faulty nodes. The concept of BFT originated from the Byzantine Generals' Problem (Lamport et al. 2019). Byzantine fault tolerance can be achieved when the functioning nodes in

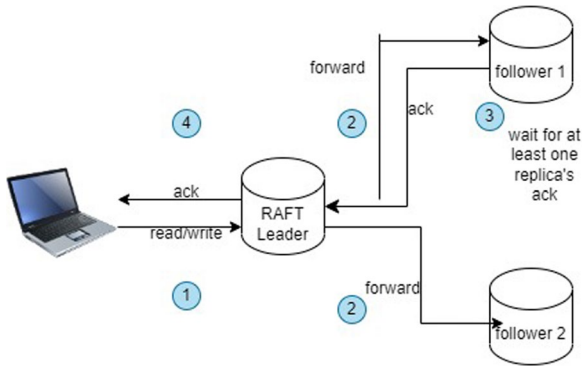


**Fig. 6** RAFT architecture

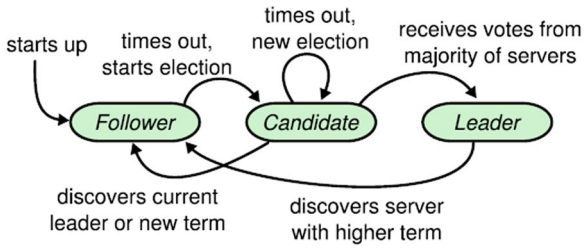


**Fig. 7** Election process

the network reach a consensus on their values. If a message is not received within a certain time limit, it can be assumed that the message from that node is faulty, and a default vote value can be assigned. Additionally, if the majority of nodes respond with a correct value, a default response can be assigned (Velliangiri and Karthikeyan 2020).

***Practical Byzantine fault-tolerant (pBFT)***

Practical Byzantine Fault Tolerance (pBFT) is a consensus algorithm developed in the late 1990s by Barbara Liskov and Miguel Castro that is optimized for asynchronous systems (Zheng and Feng 2021), where there is no upper bound on when responses to requests will be received. It was designed to have low overhead time and address issues with other Byzantine Fault Tolerance solutions. pBFT has applications in distributed computing and blockchain technology. practical Byzantine Fault Tolerance (pBFT) aims to provide a practical solution for Byzantine state machine replication in distributed systems, even when malicious nodes are present (Wang et al. 2019). In a pBFT-enabled system, nodes are sequentially ordered with one node designated as the primary (or leader) and the others as secondaries (or backups). Any eligible node in the system can become the primary by transitioning from

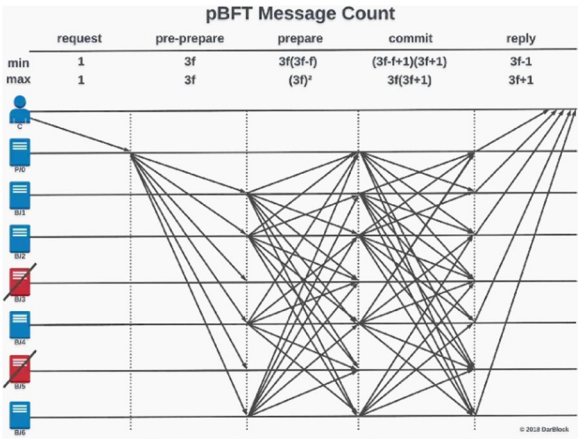


**Fig. 8** General phases of pBFT

secondary to primary, typically in the event of primary node failure. The goal of pBFT is for all honest nodes to reach a consensus about the system's state using the majority rule. A pBFT system can function as long as the maximum number of malicious nodes is less than or equal to one-third of all the nodes in the system. As the number of nodes increases, the system becomes more secure. pBFT consensus rounds are divided into four phases as shown in Fig. 8: (Wang et al. 2019)

- The client sends a request to the primary (leader) node.
- The primary (leader) node broadcasts the request to all the secondaries (backups).
- The nodes (primary and secondaries) perform the requested service and then send a reply back to the client.
- The request is successfully served when the client receives 'm+1' replies from different nodes in the network with the same result, where m is the maximum number of faulty nodes allowed.

The primary (leader) node is changed during every view (pBFT consensus round) and can be replaced by a view change protocol if a predefined amount of time has passed without the leading node broadcasting a request to the backups (secondaries). If necessary, a majority of the honest nodes can vote on the legitimacy of the current leading node and replace it with the next leading node in line.

***Proof of work (PoW)***

The process of adding a new block to the blockchain, called "mining," is performed by nodes named "miners". Miners compete to solve a complicated mathematical problem in a proof-of-work (PoW) consensus

algorithm. The first miner to solve the problem is authorized to create a new block of transactions and add it to the blockchain which is a decentralized and unchangeable record of all The process of adding a new block to the blockchain, called "mining," is performed by nodes named "miners". Miners compete to solve a complicated mathematical problem in a proof-of-work (PoW) consensus algorithm. The first miner to solve the problem is authorized to create a new block of transactions and add it to the blockchain which is a decentralized and unchangeable record of all network transactions (Fullmer and Morse 2018). The solution to the problem is authenticated by the other miners, and if it is correct the new block is added to their copy of the blockchain. Mining requires a substantial amount of computational power, and the miner who solves the problem is rewarded with a certain number of cryptocurrency units. This incentive encourages miners to participate in the process and contributes to the security of the blockchain (Gemeliarana and Sari 2018). PoW consensus algorithms are resistant to tampering and fraud because changing a block's contents would require redoing the proof-of-work for that block and all subsequent blocks, making it difficult for a single entity to control or alter the blockchain. Table 2 shows the pros and cons of PoW. In a proof-of-work (PoW) consensus algorithm, each block of transactions is linked to the previous block using a cryptographic hash value. The Process of mining is performed by "miners." A miner must select a random value (called a "nonce") and calculate the hash value of the block header, which includes the nonce and other information such as previous block hash and transaction data. If the hash value is less than a predetermined target value, the block is added to the blockchain. This process is verified by other miners in the network. The SHA-256 hash function is used in Bitcoin (Gayoso Martinez et al. 2020). By setting a target value for every 2,016 blocks, the difficulty of finding a valid hash value is maintained. Two miners may sometimes add a block at the same time; a process called "forking". In this case, all network nodes agree on the most synchronized block in the network. PoW is used in applications such as Bitcoin and Ethereum, and it takes 10 min on average to generate a block and one hour to confirm it (Vilim et al. 2016). Ethereum, in addition to being a digital currency, also serves as a platform for developing applications. The mining procedure is depicted in Fig. 9.

### *Proof of stake (PoS)*

Proof-of-Work (PoW) is a popular method for achieving distributed consensus, as seen in the Bitcoin implementation. However, PoW consumes a significant amount of energy, particularly during the Bitcoin mining process (Saad et al. 2021). A PoW system increases an entity's chances of mining a new block if it has more computational resources. Aside from the energy requirement, there are several drawbacks to using a PoW-based consensus mechanism as mentioned in table 2. A Proof-of-Stake (PoS) mechanism may be a better alternative. PoS is a type of consensus algorithm in which the next block is chosen based on the stake (amount of cryptocurrency held) of the miner (Ganesh et al. 2019), rather than their computational power. This can be a more energy-efficient way to achieve distributed consensus. In a Proof-of-Stake (PoS) consensus algorithm, nodes on a network can become candidates to validate new blocks by staking a certain amount of cryptocurrency. An algorithm then selects one of the candidates to validate the new block and earn the transaction fee. The selection algorithm uses a combination of the candidate's stake (amount of cryptocurrency held) and other factors, such as coin age and randomization, to ensure fairness among all the nodes on the network. One such factor is coin age (Nguyen et al. 2019), which tracks how long a candidate node has been a validator. The longer a node has been a validator, the higher its chances of being selected as the new validator. Another factor is random block selection, in which the validator is chosen based on a combination of the lowest hash value and the highest stake. The node with the best weighted combination of these factors becomes the new validator. Figure 10 illustrates the PoS Algorithm workflow.

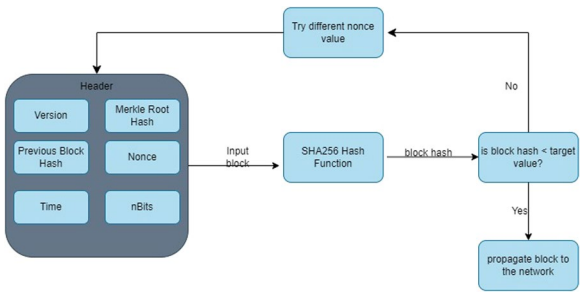


**Fig. 9** Mining process in Bitcoin network

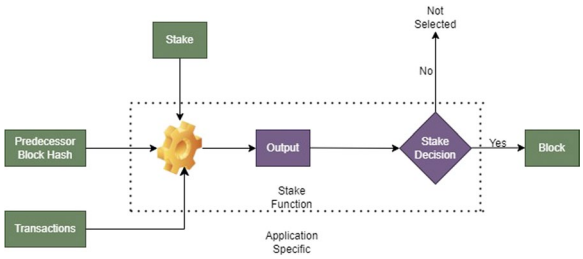


**Fig. 10** Workflow of PoS

**Table 2** Pros and Cons of PoW

| Pros | Cons |
|---|---|
| Provides a solid mechanism for achieving consensus and preventing abuses and misuses | Requires a lot of energy consumption and computational power |
| Rewards miners for securing the network and validating transactions | May lead to centralization due to mining pools and specialized hardware |
| Enables trustless transactions without intermediaries or authorities | Limits scalability due to low throughput and high latency |
| Resists attacks such as double-spending, censorship, or denial-of-service | May suffer from stagnation due to low incentives for innovation or improvement |

#### *Proof of burn (PoB)*

In Proof of Burn (PoB), validators demonstrate their commitment to the system by "burning" coins or sending them to an address from which they can never be retrieved (Karantias et al. 2020) making it unspendable. This process is used to determine which validators will be able to mine the next block in the system. Validators may burn the native currency of the blockchain application or the currency of an alternative chain, such as bitcoin, to increase their chances of being selected for block mining. Rather than investing in expensive hardware, PoB allows validators to show their long-term commitment to the system through a short-term sacrifice of coins (Karantias et al. 2020). The more currency a miner burns, the greater their chances of being selected to mine the next block on the system. The idea behind this is that by destroying their currency, the miner is showing a long-term commitment to the system and giving up a short-term gain in exchange for a potential long-term profit (Yusoff et al. 2022). To prevent early adopters from having an unfair advantage, PoB has a system in place that allows for the periodic burning of cryptocurrency to maintain mining capacity. As new blocks are mined, the energy of the burned coins decreases slightly, resulting in a deflationary process in which the overall quantity of currency decreases over time, potentially increasing its value. In contrast, cryptocurrencies that increase in quantity over time tend to lose value.

#### *Proof of capacity (PoC)*

Proof-of-Capacity (PoC) is a new mining method that is currently being used by the cryptocurrency Burstcoin (Mohamed and Ibrahim 2020). This method involves using hard disk space for mining and has the potential to be a more energy-efficient alternative to the commonly used Proof-of-Work (PoW) mining method. However, as the network has grown, mining has become increasingly difficult and energy-intensive, requiring specialized hardware known as ASICs to be effective. PoC seeks to address these issues by requiring miners to commit processing power and hard disk storage before mining begins, resulting in a faster system than PoW. PoC also has the advantage of producing blocks in four minutes as opposed to PoW's ten minutes (Mohamed and Ibrahim 2020). PoC increases miners' chances of winning the mining competition by providing more solutions, or "plots" on a computer. Overall, PoC is intended to address the energy and decentralization issues that plague PoW mining, making it a potentially appealing option for blockchain projects. Proof-of-Capacity (PoC) consists of two main components: plotting and mining (Aggarwal and Kumar 2021). Plotting involves creating a series of precomputed hashes and storing them on a hard drive using the Shabal hash function, which is used by the cryptocurrency Burstcoin (Bamakan et al. 2020). This process can take several days or weeks, depending on the size of the hard drive. The hashes are grouped into "scoops", each of which consists of two neighboring hashes. Mining entails calculating a scoop number and applying it to each nonce stored on the hard drive to determine a "deadline" value. If no one else has created a new block within that timeframe, the miner chooses the nonce with the shortest deadline and uses it to do so. If the miner creates the block before the deadline, they are rewarded with a block reward.

#### *Proof of activity (PoA)*

In Proof-of-Activity (PoA), miners utilize their computing power to solve cryptographic problems similar to Proof-of-Work (PoW) while also taking into account the amount of stake (e.g., tokens or cryptocurrency) that a miner holds, similar to Proof-of-Stake (PoS). This creates a hybrid system that combines the security of PoW with the energy efficiency of PoS (Kaur et al. 2021). By considering a miner's stake, the network can prioritize those with a long-term interest in its success rather than just those with the most powerful computing resources. PoA can be an effective way to balance security and efficiency in a blockchain, but it may also be more complex to implement and potentially less secure compared to pure PoW or PoS systems. In Proof-of-Activity (PoA), the mining process begins like a Proof-of-Work (PoW) process, with miners using their computing power to solve mathematical equations and create new blocks. When a new block is successfully mined, the system transitions to a Proof-of-Stake (PoS) phase. A group of validators

is randomly selected to sign the new block (Andola et al. 2020), which is validated based on the details in its header. Validators with a larger amount of cryptocurrency have a higher chance of being chosen as signers. If the required number of validators sign the new block, it is considered complete and added to the existing blockchain, with the transactions in the block being recorded. If the selected signers are not present to sign the new block, the process moves to the next winning block, where a new group of validators is chosen based on their cryptocurrency holdings. If a winning block does not receive the required number of signatures to become complete, the process continues. The first miner and any validators who contributed to the new block are rewarded. PoA is criticized for its partial use of PoW and PoS, but it also prevents the risk of a 51% attack (Shrimali and Patel 2022; Sayeed et al. 2019).

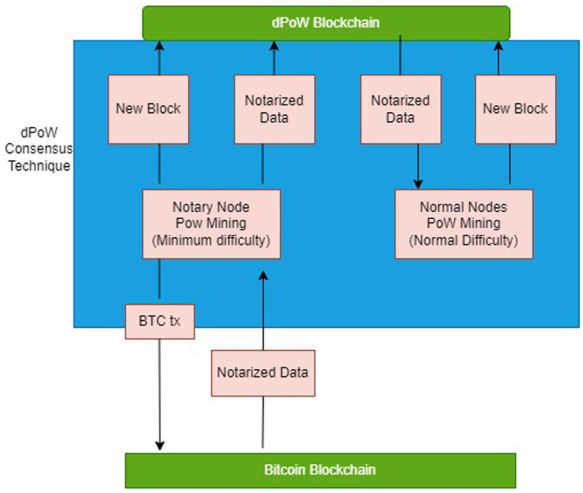


**Fig. 11** dPoW consensus mechanism

#### *Delayed proof of work (dPoW)*

Delayed Proof of Work (dPoW) is a consensus algorithm that aims to improve the security of a blockchain network by incorporating elements from a more secure blockchain. This is achieved by allowing a secondary blockchain, known as the "notary chain," to record hashes of blocks from the primary blockchain, known as the "target chain". In a dPoW system, miners on the notary chain compete to create new blocks just like in a traditional Proof-of-Work (PoW) system. However, instead of including transactions in these blocks, the miners include the hashes of blocks from the target chain. This process is known as "notarization". The notarized blocks are then added to the notary chain,

providing an additional layer of security for the target chain (Osadchuk and Oliynykov 2019). If an attacker attempted to change the transactions on the target chain, they would also have to change the corresponding notarized block on the notary chain, which would be much more difficult due to the notary chain's increased security. This makes it much more difficult for an attacker to successfully alter the target chain, increasing its security (Sayeed et al. 2020). dPoW can be a good way to increase the security of a blockchain network, especially for smaller or less secure networks that are more vulnerable to attacks. It does necessitate the use of a secondary, more secure blockchain, which adds complexity and may not be practical in all situations. Figure 11 depicts the algorithm.

#### *Delegate proof of stake (DPoS)*

In Delegated Proof of Stake (DPoS), token holders (stakers) can assign their voting power to delegates or witnesses to create new blocks and validate transactions on the blockchain (Yang et al. 2019).

These delegates are chosen by the stakers through a voting process, and the number of votes a delegate receives is determined by the number of tokens they hold. The delegates with the most votes are responsible for creating new blocks and validating transactions, while the remaining delegates act as backups (Saad et al. 2020; Bachani and Bhattacharjya 2022). DPoS is intended to be more efficient and scalable than other consensus algorithms such as Proof-of-Work (PoW) or Proof-of-Stake (PoS), as it allows a small group of delegates to handle the majority of the network's transactions. It also gives token holders more influence over the network, as they can vote for or against specific delegates. However, DPoS can also be more centralized as the power to create new blocks and validate transactions is held by a small group of delegates, making it susceptible to collusion and manipulation (Hu et al. 2021). Table 3 shows the pros and cons for each PoX consensus algorithm. Table 3 compares the different types of PoX consensus algorithm in terms of advantages and disadvantages, and we can conclude form Table 3 that the best consensus algorithm depends on the situation and the predefined constraints of the system itself. Table 4 shows the advantages and disadvantages of different PoX algorithms in detail in terms of energy efficiency, throughput, scalability, and security. Table 3 and Table 4 provide an overview of commonly utilized consensus algorithms in blockchain networks. Each algorithm has unique advantages and disadvantages that should be taken into consideration when selecting an appropriate consensus algorithm for a particular blockchain project. Proof of Work (PoW) is a well-known consensus algorithm that offers substantial protection against attacks but requires significant energy consumption and computational resources. On the other hand, Proof of Stake

**Table 3** Pros and cons for each PoX consensus algorithm

| Consensus algorithms | References | Pros | Cons |
|---|---|---|---|
| Proof of Work (PoW) | (Yang et al. 2019; Puthal and Mohanty 2018; Dey 2018) | Protect against DoS attacks and spam | Intensive usage of energy and resources required for the system |
| | | Secure the entire network | The risk of a 51% attack |
| | | Prevent double spending attacks in distributed systems | |
| Proof of Stake (PoS) | (Vashchuck and Shuwar 2018; Nguyen et al. 2019; Yang et al.2019) | Less power consumption and less hardware usage compared to PoW | Coin hoarding |
| | | | Monopolization |
| | | | Double spending |
| Proof of Burn (PoB) | (Karntias et al. 2020; Menon et al. 2022) | Reduced reliance on computational resources | The destruction of coins leads to resource waste |
| | | User commitment over the long term | Hoarding of coins |
| | | Decentralized structure | Attempts to manipulate the system through control of a large number of coins |
| | | Lower energy consumption compared to proof-of-work systems | |
| Proof of Activity (PoA) | (Salimitari et al. 2020; Wang et al. 2020) | This system is more secure against 51% attacks than PoS and PoW | Intensive use of resources is required to carry out certain actions |
| | | It is resistant to DoS attacks | The possibility of a monopoly forming increases if there is a high risk of penalty for attempting to double sign transactions |
| | | It promotes decentralization | There is a balance to be struck between the energy needed to perform certain actions and the potential for monopoly power |
| Proof of Space (PoS) | (Park et al. 2018; Benisi et al. 2020) | Energy efficiency due to the use of low-power hard drives instead of specialized hardware such as ASICs, CPUs, and GPUs | Verification efficiency and storage availability are challenging task |
| | | Greater potential for decentralization | The potential for a probabilistic monopoly with large amounts of space |
| Delegated Proof of Stake (DPoS) | (Saad et al. 2020; Do et al. 2019) | Speed: DPoS can facilitate faster transaction processing and block production compared to other proof of stake algorithms | Potential for centralization: The use of delegates in DPoS can potentially lead to centralization if the same small group of delegates are consistently elected to represent the network |
| | | Energy efficiency: DPoS uses significantly less energy than proof of work algorithms, making it more environmentally friendly | Limited participation: Only those with a significant number of tokens can participate in the delegate selection process, which may exclude some members of the community |
| | | Decentralization: DPoS allows for a more decentralized network by allowing token holders to vote for "delegates" who will represent them in the decision-making process | Complexity: DPoS is a more complex system than traditional proof of stake algorithms, which may make it more difficult to understand and implement |
| Delayed Proof of Work (dPoW) | (Sayeed et al. 2019) | Increased security: DPoW uses a secondary blockchain to secure the main chain, providing an additional layer of protection against 51% attacks | Complexity: DPoW is a more complex system than traditional proof of work algorithms, which can make it more difficult to understand and implement |
| | | Decentralization: DPoW can help to decentralize the mining process by allowing a wider range of miners to participate in the network | Dependency on secondary chain: DPoW relies on a secondary chain to secure the main chain, which means that if the secondary chain becomes compromised, the main chain may also be at risk |
| | | Energy efficiency: DPoW uses less energy than traditional proof of work algorithms, making it more environmentally friendly | Compatibility issues: DPoW may not be compatible with certain types of software or hardware, which could limit its use in certain situations |

**Table 4** Pros and cons for each PoX consensus algorithm in terms of various criteria

| Algorithm | Energy efficiency | Throughput | Scalability | Security |
|---|---|---|---|---|
| PoW | Low | Low | Low | High |
| PoS | High | Medium | Medium | Medium |
| PoB | High | Medium | Medium | Medium |
| PoA | Medium | High | High | High |
| PoS (space) | High | Medium | High | Low |
| DPoS | High | Very high | Very high | Low to medium |
| dPoW | High | High | High | High |

(PoS) presents an alternative approach that consumes less energy but may be susceptible to coin hoarding and monopolization.

Proof of Burn (PoB) is a relatively new consensus algorithm that prioritizes user commitment over the long term. However, coin destruction leads to resource waste, and coin hoarding can manipulate the system. Proof of Space (PoS) is energy-efficient and highly decentralized, using low-power hard drives, but it may be vulnerable to monopolization with a large amount of space. Proof of Activity (PoA) is a recently developed consensus algorithm that resists DoS attacks and encourages decentralization, but certain actions require resources. Delayed Proof of Work (dPoW) adds an extra layer of protection against 51% attacks, and it is more energy efficient than traditional PoW algorithms. However, it is more complex and has dependency issues. Delegated Proof of Stake (DPoS) is a consensus algorithm that allows token holders to vote for delegates who represent them in the decision-making process. DPoS is fast and energy-efficient, but the use of delegates may lead to centralization and exclude some members of the community from participating. In general, there is no one-size-fits-all solution for selecting a consensus algorithm. The choice of algorithm will depend on the specific needs and goals of the blockchain project.

#### DAG-based consensus algorithms

DAG is a distributed ledger technology that is built on the principles of directed acyclic graphs (DAGs). These algorithms portray transactions as nodes within the DAG, and the edges between nodes display the interdependence between the transactions (Chen et al. 2018). One of the main advantages of DAG-based consensus algorithms over conventional blockchain-based consensus algorithms is their capacity to process transactions more quickly and flexibly. This is because DAG-based algorithms do not contain blocks that must be added to the chain in a particular order and do not require miners to do expensive proof-of-work calculations. Instead, transactions are added to the DAG in parallel, resulting in higher throughput. Examples of DAG-based consensus algorithms include IOTA's Tangle, Nano's Block Lattice, and Hashgraph. These algorithms have been used in a variety of applications, like distributed ledger technologies, peer-to-peer networks, and decentralized applications. DAG-based consensus algorithms are still a relatively new and rapidly evolving field of technology, and there is ongoing debate about their relative advantages and disadvantages compared to traditional blockchain-based consensus algorithms. IOTA is a distributed ledger technology with a primary structure based on a directed acyclic graph (DAG). It was designed to help the Internet of Things (IoT) (Silvano and Marcelino 2020), which is a network of interconnected devices that can interact and transfer data. One of IOTA's key characteristics is its scalability, which is achieved through the use of the Tangle, a DAG-based consensus algorithm. Unlike traditional blockchain-based systems that rely on miners to perform proof-of-work calculations to validate transactions, IOTA employs a different approach known as "proof-of-workless" consensus. In this approach, each transaction in the Tangle must validate two other transactions before it can be added to the DAG. This allows IOTA to achieve a high level of throughput and low transaction fees, making it suitable for use in the IoT. IOTA also utilizes a unique form of cryptocurrency called MIOTA, which is used to facilitate transactions on the IOTA network. In addition to being used as a means of exchange, MIOTA can also be used to represent data or store values (Lamberti et al. 2019). IOTA has been used in various applications, including supply chain management, smart cities, and energy markets. However, it's worth noting that IOTA has faced some controversy and criticism in the past, including concerns about the security and centralization of its network. Nano is a cryptocurrency that utilizes a directed acyclic graph (DAG)-based consensus algorithm called the Block Lattice (Morais et al. 2020). In the Block Lattice, each Nano account has its own blockchain, called an "account-chain," which is used to track the balance and transaction history of the account. The Block Lattice's scalability is one of its main benefits, as transactions can be processed in parallel due to each account having its own blockchain, allowing for a high level of throughput. The Block Lattice also uses a proof-of-workless consensus algorithm, meaning that transactions are validated using a voting process instead of the costly proof-of-work calculations used by conventional blockchain based systems. Nano has several other features that make it appropriate for use

as a cryptocurrency, including fast transaction times, low transaction fees, and energy efficiency. It has been used in various applications such as peer-to-peer payments, online micropayments, and online gaming. It is important to note that Nano has faced some controversy and criticism in the past, including concerns about the centralization of its network and the security of its consensus algorithm. Nano team has continued to work on improving the technology and addressing these issues.

#### *Jointgraph*

Jointgraph is a consensus algorithm that is based on Byzantine fault tolerance. It uses events to pack transactions and sends them through a gossip protocol, which allows anyone to send events to a random node (Xiang et al. 2021). These events are validated by all members of the network, and Jointgraph uses a threshold of 2/3 of all members to reach consensus. To improve consensus efficiency, Jointgraph employs a supervisor node that monitors member behavior and collects votes during the consensus process to determine the finality of events. Every member of the network has a copy of Jointgraph, so it is possible to know each member would vote in the consensus process. An example of Jointgraph consensus is illustrated in Fig. 12, which shows three ordinary nodes (A, B, and C) and one supervisory node (D). Red circles represent events that have reached consensus, while light circles represent events that are not in consensus. Red circles are events that are verified by at least three members, including the supervisory node. The confirmation time for events depends on the frequency of the gossip protocol.

#### *BlockDAG*

BlockDAG is a DAG-based consensus algorithm that uses sorting and merging to reconstruct a single-chain-based blockchain system (Gai et al. 2020). It has five phases: Block Generation (BG), Sorting Block (SB), Block Merging (BM), Consensus Implementation (CI), and Block Splits (BS). In the first phase, BG generates original blocks by adding transactions to the block pool of the nearest blockchain node, which then validates the transactions and bundles them into blocks that are added to the system block pool for further validation. In the second phase, SB sorts all unvalidated blocks in the block pool using a sorting algorithm to create a sequence in the DAG structure. BlockDAG addresses issues of double spending and consensus conflicts in the BM phase, where the merged blocks are finalized through global consensus in the CI phase. In the BS phase, the merged blocks are split into their original states and placed in the on-premises BlockDAG structure (Gai et al. 2020).

#### *UL-BlockDAG*

UL(Unsupervised Learning)-blockDAG is a distributed ledger system that utilizes a directed acyclic graph (DAG) as its primary data structure. It is an extension of the blockDAG consensus algorithm, which merges and arranges data to reconstruct a blockchain system based on a single chain. The system's key feature is its UL-scalability, which enables it to handle many transactions per second (TPS) with a high throughput rate. In addition, it uses a proof-of-workless consensus method, certifying transactions through voting rather than the costly proof-of-work calculations used in traditional blockchain systems. UL-blockDAG also offers other features such as low transaction fees, fast transaction times, and energy efficiency, making it ideal for use in distributed ledger technologies and decentralized applications. UL-blockDAG has already been employed in various areas, including supply chain management, smart cities, and energy markets. However, it is a relatively new and rapidly evolving technology, and there is ongoing debate about its relative advantages and disadvantages compared to other distributed ledger technologies. (Reddy and Sharma 2020)

#### *Dexon*

DEXON is a distributed ledger technology that uses a consensus algorithm based on proof of participation (PoP). In DEXON, every node is equally likely to propose a block, and the issuer of a block is determined using a Verifiable Random Function (VRF) (Chen et al. 2018). This reduces communication costs and encourages more nodes to join the protocol. DEXON uses a block lattice structure and proposes the use of a fast Byzantine agreement that terminates in $6\sigma$ time, where $\sigma$ is the upper bound of the network's gossip period. It generates on-chain, unpredictable randomness as it achieves consensus, and once a DEXON Byzantine agreement confirms a block, a committee of nodes generates a threshold signature with an unpredictable threshold value. In this consensus, no single block proposer can determine the consensus timestamp of a proposed block, and DEXON achieves second-level latency instead of traditional minute-level latency. DEXON is highly decentralized and robust in practical deployment environments.

#### *Spectre*

SPECTRE is a protocol for the consensus core of cryptocurrencies that offers high throughput and fast confirmation times (Kovalchuck et al. 2022). It enables high block creation rates and uses partial synchronous networks. SPECTRE generalizes Nakamoto's blockchain into a block DAG, allowing miners to create blocks concurrently by maintaining a full DAG of blocks. It

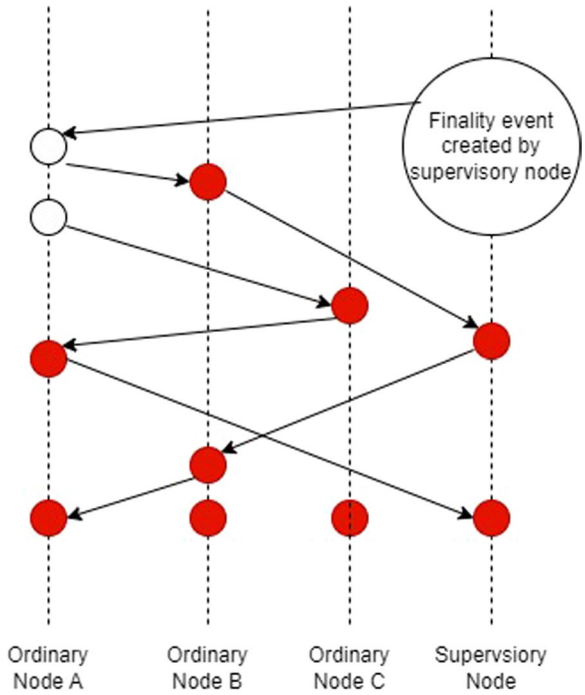


**Fig. 12** Jointgraph consensus algorithm

uses a voting algorithm to determine the order between each pair of blocks in the DAG, with the votes coming from blocks rather than miners. The votes are algorithmically determined based on the location of the block within the DAG, and the majority vote becomes irreversible very quickly, providing a consistent set of transactions. Tables 5 and 6 Show the analysis of DAG based consensus algorithms in terms of pros and cons, energy efficiency, throughput, scalability and security. The table summarizes different consensus algorithms along with their references, pros, and cons. The consensus algorithms discussed are Jointgraph, BlockDag, UL-blockDAG, Dexon, and Spectre. Jointgraph is designed to be robust against double spending attacks and has improved throughput compared to other distributed ledger technologies. BlockDag has high scalability, robustness, and high throughput but may increase latency when using a merge sort. UL-blockDAG has robustness, high transaction rate, and a high block creation rate but the complexity of the system increases as the number of nodes increases. Dexon has low latency and high throughput, reduced cost, scalability, and usefulness in resource-constrained environments. Spectre is robust and has a high transaction rate and block creation rate but has a high latency and double spending risk. This section examined the use of directed acyclic graphs (DAGs) as an alternative to conventional blockchain-based consensus algorithms. DAG-based consensus algorithms, such as IOTA's Tangle, Nano's Block Lattice, Jointgraph, BlockDAG, and UL-BlockDAG, have the ability to process transactions quickly and flexibly by adding them to the DAG in parallel, leading to a higher throughput. Unlike traditional blockchain-based consensus algorithms, DAG-based algorithms do not require miners to perform costly proof-of-work calculations and do not contain blocks that must be put on the chain in a specific order. The passage cites examples of applications that have utilized DAG-based consensus algorithms, including distributed ledger technologies, peer-to-peer networks, and decentralized applications. However, there are ongoing debates regarding the advantages and disadvantages of DAG-based consensus algorithms compared to traditional blockchain-based consensus algorithms. The passage provides details on IOTA's Tangle and Nano's Block Lattice as DAG-based consensus algorithms, highlighting their unique features and criticisms. Additionally, the passage briefly explains other DAG-based consensus algorithms such as Jointgraph, BlockDAG, and UL-BlockDAG.

#### Hybrid-based consensus algorithms

Hybrid consensus algorithms combine elements from various types of consensus mechanisms to achieve specific properties or goals (Wu et al. 2020). These algorithms can adapt to changing conditions or requirements by using multiple mechanisms in parallel or sequentially. They can also strike a balance between decentralization and efficiency, allowing for faster transaction processing while remaining decentralized. Delegated Proof of Stake (DPoS), Hybrid Proof of Work/Proof of Stake (PoW/PoS), and Byzantine Fault Tolerance are examples of hybrid consensus algorithms (BFT). These algorithms seek to combine the advantages of various consensus mechanisms to create a more flexible, adaptable, and secure system. One example of a hybrid consensus algorithm is a proposed algorithm that improves the efficiency and scalability of conversation interactions in multidomain systems, where the number of conversations increases rapidly with the number of domains (Yang et al. 2022). The proposed algorithm combines reputation-driven voting, dynamic construction strategies, and an incentive scheme with both economic and non-economic rewards to create a decentralized, scalable, and secure system for experts to communicate across domains. The researchers built a master–slave chain model for a multidomain conversation system that can process multiple interactions concurrently. They tested the proposed approach on a local private blockchain network and showed that

**Table 5** Pros and cons for DAG based consensus algorithms

| Consensus algorithms | References | Pros | Cons |
|---|---|---|---|
| Jointgraph | (Xiang et al. 2021) | The consensus algorithm is designed to be robust against double spending attacks, which can help ensure the integrity of the network and prevent fraud. JointGraph is also known to have improved throughput compared to other distributed ledger technologies, such as hashgraph, which means it can handle a higher volume of transactions in a given amount of time | A central point of control in a network or system can be a target for attack |
| BlockDAG | (Gai et al. 2020) | High scalability, robustness, and high throughput | Parallel processing requires the assurance of transaction records only once, but this may increase latency when using a merge sort. The process of splitting transactions for parallel processing can also be vulnerable to attack |
| UL-BlockDAG | (Reddy and Sharma 2020) | Robustness, high transaction rate, and block creation rate is high | As the number of nodes increases, the complexity of the system also increases |
| Dexon | (Chen et al. 2018) | low latency and high throughput include reduced cost, transaction ordering fairness, scalability, unpredictable randomness, secure transaction finality, and usefulness in resource-constrained environments | Under research |
| Spectre | (Kovalchuck et al. 2022) | Robustness, high transaction rate, and block creation rate is high | High latency, double spending risk |

**Table 6** Pros and cons for DAG based consensus algorithms based on different criteria

| Algorithm | Energy efficiency | Throughput | Scalability | Security |
|---|---|---|---|---|
| Jointgraph | High | High | High | High |
| BlockDAG | Medium | Medium | Medium | Medium |
| UL-BlockDAG | Medium | Medium | Medium | High |
| Dexon | High | High | High | High |
| Spectre | Low | High | High | High |

it was feasible and effective in offering secure, decentralized, and scalable multidomain conversation interactions. Another example is EOS, a well-known blockchain software system with the highest market value aside from Bitcoin and Ethereum (Rahman and Mohsin 2020). It uses a consensus mechanism called BFT-DPoS, which is a hybrid of delegated proof of stake and Byzantine fault tolerance. In the consensus process, nodes are voted on to determine decision makers through the DPoS algorithm, and then these decision makers communicate with each other to form the block sequence of the system. This results in the continuous generation of six blocks every 0.5 s, minimizing the delay in block propagation, increasing the speed of block generation, and greatly increasing the number of transactions. This allows EOS to support a customer base of millions using blockchain technology.

This section discussed hybrid consensus algorithms, which combine elements from different types of consensus mechanisms to achieve specific properties or goals. These algorithms strike a balance between decentralization and efficiency, allowing for faster transaction processing while remaining decentralized. Examples of hybrid consensus algorithms include Delegated Proof of Stake (DPoS), Hybrid Proof of Work/Proof of Stake (PoW/PoS), and Byzantine Fault Tolerance (BFT). The passage also provides examples of two hybrid consensus algorithms, one proposed algorithm that improves the efficiency and scalability of conversation interactions in multidomain systems, and another is EOS, a well-known blockchain software system that uses a consensus mechanism called BFT-DPoS.

## Future improvements to consensus algorithms

It is uncertain what the future holds for consensus algorithms, as the field is constantly changing, and new technologies are being introduced. However, some trends that may impact the future of consensus algorithms include an emphasis on increased efficiency to reduce resource intensity and energy consumption, a focus on enhancing security to protect against vulnerabilities and attacks, wider adoption in a variety of industries and applications, and the need for scalability to handle a growing volume of transactions.

### Traditional consensus algorithm improvements

Traditional consensus algorithms frequently have flaws and limitations. Researchers have been working to refine and improve the original algorithms to focus on these issues and improve them. While maintaining the benefits of the algorithms. The aim is to address and overcome their weaknesses in order to expand the development and evolution of consensus algorithms. The Proof-of-Work (PoW) algorithm has several shortcomings, including slow consensus formation, low data throughput, and high computing power utilization. There are numerous ways to improve and enhance the PoW algorithm. The low data throughput of the blockchain can be improved by raising the block size and decreasing the block creation interval. Key blocks are used for leader elections but do not contain transaction information, while micro-blocks are used to hold transaction information according to the Bitcoin-NG protocol (Eyal et al. 2016). As a result, block generation can be done more quickly and effectively while using less computational power. The Ethash algorithm addresses the issue of high computing power consumption by introducing I/O blocking and a directed acyclic graph to improve the target value solution in PoW. It uses a small dataset to verify block information and a large dataset generated from the small dataset for mining, and miners can only save the large dataset to mine more efficiently. The Ethash algorithm also uses dynamic adjustment to improve the production speed of data blocks and reduce transaction times. It is designed to be more suitable for general-purpose computers with large memory capacities rather than requiring specific hardware. The Proof-of-Stake (PoS) algorithm suffers from centralization issues due to the exclusive accounting rights held by high-stakes nodes. Researchers have tried to focus on these problems across various approaches. One of them involves imitating the Proof-of-Work (PoW) algorithm by using virtual mining technology, which requires only a small amount of computing resources for contributing nodes. This eliminates the competition for computing power and ensures randomness in the selection of new blocks, while also reducing the risk of centralization, avoiding the waste of computing power, and increasing fairness in mining. Another approach includes mixing the PoS algorithm with the Byzantine fault-tolerant algorithm. By giving varying weights to votes based on stake and requiring a weight of more than two-thirds of the total weight to attain consensus, the Algorand and Ethereum protocols, also known as the BA protocol and the Casper Friendly Finality Gadget protocol, employ this approach. An illustration of this kind of development is the Ouroboros algorithm employed in the ADA coin system. Additionally, it offers a compensation system to encourage trustworthy nodes to donate efficient processing power and move each node's behavior closer to a Nash equilibrium, where shifting a node's approach does not boost their own profitability. ThePoS algorithm's security may be enhanced by this. The effectiveness of an algorithm is largely determined by its efficiency. A good consensus algorithm should be efficient, have a low delay, be secure, and be stable. While traditional consensus algorithms can ensure the smooth operation of a blockchain system, they often lack efficiency. For example, the Proof of Work (PoW) consensus algorithm can take around ten minutes for each hash calculation and has a confirmation delay of approximately one hour. This level of efficiency is not sufficient for practical use in cases where frequent computations are needed. The efficiency of an algorithm plays a significant role in the adoption and implementation of blockchain in real-world situations, and improving the efficiency of consensus algorithms is an important area of development in the field. One example of an optimized algorithm is the Matrix Proof of Work (MPoW), which is based on the PoW algorithm and uses matrix calculations to reduce the block time (Zeng et al. 2019). Another example is the Proof of Trust (PoT) algorithm, which introduces a trust-proof mechanism to dynamically assign trust to nodes in the blockchain, with higher trust leading to a higher probability of accounting. This algorithm reduces network delay and the time required for consensus, improving the overall efficiency of the system. In the next section, we are going to support our review with a table that characterizes the different applications of consensus algorithms and which technologies utilize a certain consensus algorithm. Table 7 depicts some examples of the different technologies and various domains in which the consensus algorithms are being used.

## Conclusion

This topic has been mentioned many times in other papers, and many researchers have made a great effort, like (Zheng et al. 2017) who explained the architecture of blockchain. The paper also reviewed some of the existing consensus algorithms used on different blockchain platforms, such as proof-of-work (PoW), proof-of-stake (PoS), practical Byzantine fault tolerance (PBFT), delegated proof-of-stake (DPoS), proof-of-elapsed-time (PoET), and proof-of-authority (PoA). The paper compares these algorithms based on their performance, security, scalability, and energy efficiency. In addition to some future perspectives.

In (Xu et al. 2019b) the authors provided a systematic review of blockchain literature from various disciplines and perspectives. The article analyzes 41 papers published in Web of Science (WOS) from 2016 to 2019 that cover different aspects of blockchain, such as its definition, characteristics, classification, applications, challenges, and future directions. The article finds that there is no consensus on the definition of blockchain among researchers, but most of them agree that it has some key features such as decentralization, immutability, consensus mechanisms, cryptography, and smart contracts. The article also proposes a classification scheme for blockchain based on its architecture (public vs. private), governance (permissionless vs. permissioned), and functionality (generic vs. specific). The article reviewed some of the existing and potential applications of blockchain in various domains such as finance, supply chain management, healthcare, education, energy, government, and social media. The article identifies some of the benefits and challenges of blockchain adoption in these domains, such as efficiency improvement, cost reduction, trust enhancement, security enhancement, privacy protection, scalability issues, regulatory issues, interoperability issues, and user acceptance issues. In our work, unlike (Zheng et al. 2017) and (Xu et al. 2019b) we have focused on the inclusivity of the review, so we have included most of the aspects of the consensus algorithms, the old school, and the new trends in the industry. We have also compared the performance of every one of them. Moreover, we have included the applications that are utilizing these consensus algorithms. This work is considered inclusive and covers different aspects of the industry without skipping the basics of the field. In conclusion, consensus algorithms are an essential component of decentralized systems and have several applications in distributed databases, distributed ledgers, and blockchain technology. Several well-known consensus algorithms, including Proof of Work, Proof of Stake, and Practical Byzantine Fault Tolerance, have been featured in our review. These algorithms are useful for various use cases since they each have distinctive characteristics and trade-offs. For instance, Proof of Work is frequently used in blockchain technology and offers a high level of security and decentralization, but it also has scalability problems and uses a lot of energy. In contrast, Proof of Stake uses a different consensus technique to address the scalability and energy consumption problems associated with Proof of Work. It also has its own set of drawbacks, including the Practical Byzantine Fault Tolerance, while being a more mature and widely adopted algorithm, is less commonly used in blockchain technology, due to its requirement of a high number of confirmations before a block can be added to the chain. The application of a consensus algorithm also has a substantial impact on the system's performance and security. It is vital to assess the algorithm's applicability for a particular use case and take into account a number of variables, including scalability, security and energy consumption. Despite advancements in the field, scalability and security issues in decentralized systems continue to pose significant challenges. To address these challenges continuing research is needed to enhance the scalability and security of existing consensus algorithms and develop new mechanisms that can more effectively address these issues. Moreover, the environmental impact of consensus algorithms like energy consumption and sustainability must also be taken into consideration. In the future, there is a growing interest in researching new consensus algorithms for distributed ledger technologies, including those based on sharding, hybrid algorithms that combine multiple consensus mechanisms, methods for reducing energy consumption in PoW algorithms and addressing security issues in PoS algorithms. In summary, consensus algorithms are critical for the functioning of decentralized systems, and ongoing research and development in this field is crucial for the advancement and widespread adoption of decentralized technologies. The selection of the appropriate consensus algorithm for a given use case is a critical decision that

**Table 7** Different domains and applications with different consensus algorithms

| Domain | Application | Consensus algorithm |
|---|---|---|
| DBMS | Google Megastore | PAXOS |
| | Apache cudu | RAFT |
| | Cockroach DB | RAFT |
| Cryptocurrency | Bitcoin | PoW |
| | Etherum | PoW,PoS |
| | SlimCoin | PoB |
| | BurstCoin | PoC |
| | IOTA | FPC |
| | EOS | DPoS |
| | neo | DBFT |
| | Vet | PoA |
| | NXT | PoS |
| Platforms for developments | Microsoft Azure | PoA |
| | GoChain | PoR (Reputation) |
| | Hyperledger | PBFT |
| E-commerce | ALgorand | PoS |
| | BitShare | DPoS |
| Health Care | eHealth Estonia | Pow, pBFT |
| | Farma Trust | Etherum(PoS, PoW) |
| Supply Chain | EverLEdger | PoA,PoS |

can greatly impact the performance and security of the system. Further research is needed to improve scalability and security of these algorithms and to develop new mechanisms that can more effectively address the challenges of decentralized systems. The consensus algorithm is an active area of research with great potential, and the future of distributed systems holds promise for new and innovative consensus mechanisms.

**Acknowledgements**
The authors would like to thank the editor and anonymous referees.

**Authors' contributions**
All the authors read and approved the final manuscript.

**Funding**
Not applicable.

**Availability of data and materials**
Not applicable.

## Declarations

**Competing interests**
The authors declare no competing interests.

Received: 23 January 2023 Accepted: 31 May 2023
Published online: 03 November 2023

**References**

Aggarwal S, Kumar N (2021) Cryptographic consensus mechanisms. In: Advances in computers. Elsevier, vol 121, pp 211–226
Ammous S (2016) Blockchain technology: what is it good for? SSRN 2832751
Andola N, Venkatesan S, Verma S et al (2020) PoEWAL: a lightweight consensus mechanism for blockchain in IoT. Pervasive Mob Comput 69:101291
Andrey A, Petr C (2019) Review of existing consensus algorithms blockchain. In: 2019 international conference "quality management, transport and information security, information technologies" (IT &QM &IS). IEEE, pp 124–127
Bachani V, Bhattacharjya A (2022) Preferential delegated proof of stake (PDPoS)-modified DPoS with two layers towards scalability and higher TPS. Symmetry 15(1):4
Baldominos A, Saez Y (2019) Coin.AI: a proof-of-useful-work scheme for blockchain-based distributed deep learning. Entropy 21(8):723
Bamakan SMH, Motavali A, Bondarti AB (2020) A survey of blockchain consensus algorithms performance evaluation criteria. Expert Syst Appl 154:113385
Benisi NZ, Aminian M, Javadi B (2020) Blockchain-based decentralized storage networks: a survey. J Netw Comput Appl 162:102656
Bentov I, Lee C, Mizrahi A, Rosenfeld M (2014) Proof of activity: extending bitcoin's proof of work via proof of stake [extended abstract]. ACM SIGMETRICS Perform Eval Rev 42(3):34–37
Chen T-Y, Huang W-N, Kuo P-C, Chung H, Chao T-W (2018) DEXON: a highly scalable, decentralized DAG-based consensus algorithm. arXiv preprint arXiv:1811.07525
De Prisco R, Lampson B, Lynch N (2000) Revisiting the Paxos algorithm. Theoret Comput Sci 243(1–2):35–91
Denisova V (2019) Blockchain infrastructure and growth of global power consumption. Int J Energy Econ Policy
Dey S (2018) Securing majority-attack in blockchain using machine learning and algorithmic game theory: a proof of work. In: 2018 10th computer science and electronic engineering (CEEC). IEEE, pp 7–10
Do T, Nguyen T, Pham H (2019) Delegated proof of reputation: a novel blockchain consensus. In: Proceedings of the 1st international electronics communication conference, pp 90–98
El Ioini N, Pahl C (2018) A Review of distributed ledger technologies: confederated international conferences: CoopIS, C &TC, and ODBASE 2018, Valletta, Malta, October 22–26, 2018. Proceedings, Part II:277–288. https://doi.org/10.1007/978-3-030-02671-4_16
Eyal I, Gencer AE, Sirer EG, Van Renesse R (2016) Bitcoin-ng: a scalable blockchain protocol. In: 13th {USENIX} symposium on networked systems design and implementation ({NSDI} 16), pp 45–59
Fu X, Wang H, Shi P (2021) A survey of blockchain consensus algorithms: mechanism, design and applications. Sci China Inf Sci 64:1–15
Fullmer D, Morse AS (2018) Analysis of difficulty control in bitcoin and proof-of-work blockchains. In: 2018 IEEE conference on decision and control (CDC). IEEE, pp 5988–5992
Gai K, Hu Z, Zhu L, Wang R, Zhang Z (2020) Blockchain meets DAG: a BlockDAG consensus mechanism. In: Algorithms and architectures for parallel processing: 20th international conference, ICA3PP 2020, New York City, NY, USA, October 2–4, 2020, Proceedings, Part III, vol 20. Springer, pp 110–125
Ganesh C, Orlandi C, Tschudi D (2019) Proof-of-stake protocols for privacy-aware blockchains. In: Advances in cryptology—EUROCRYPT 2019: 38th annual international conference on the theory and applications of cryptographic techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I. Springer, vol 38, pp 690–719
Gayoso Martinez V, Hernández-Álvarez L, Hernandez Encinas L (2020) Analysis of the cryptographic tools for blockchain and bitcoin. Mathematics 8(1):131
Gemeliarana IGAK, Sari RF (2018) Evaluation of proof of work (POW) blockchains security network on selfish mining. In: 2018 international seminar on research of information technology and intelligent systems (ISRITI). IEEE, pp 126–130
Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S (2016) On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 3–16
Guo H, Yu X (2022) A survey on blockchain technology and its security. Blockchain Res Appl 3(2):100067
Han T, Gao K (2020) Review of blockchain consensus algorithms. Sci J Intell Syst Res 2(12)
Hu J, Liu K (2020) Raft consensus mechanism and the applications. J Phys Conf Ser 1544:012079
Hu Q, Yan B, Han Y, Yu J (2021) An improved delegated proof of stake consensus algorithm. Procedia Comput Sci 187:341–346
Ismail L, Materwala H (2019) A review of blockchain architecture and consensus protocols: use cases, challenges, and solutions. Symmetry 11(10):1198
Karantias K, Kiayias A, Zindros D (2020) Proof-of-burn. In: Financial cryptography and data security: 24th international conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 revised selected papers 24. Springer, pp 523–540
Kaur M, Khan MZ, Gupta S, Noorwali A, Chakraborty C, Pani SK (2021) MBCP: performance analysis of large scale mainstream blockchain consensus protocols. IEEE Access 9:80931–80944
Kim D, Doh I, Chae K (2021) Improved raft algorithm exploiting federated learning for private blockchain performance enhancement. In: 2021 international conference on information networking (ICOIN). IEEE, pp 828–832
Kovalchuk L, Oliynykov R, Bespalov Y, Rodinko M (2022) Comparative analysis of consensus algorithms using a directed acyclic graph instead of a blockchain, and the construction of security estimates of spectre protocol against double spend attack. In: Information security technologies in the decentralized distributed networks. Springer, pp 203–224
Lamberti R, Fries C, Lücking M, Manke R, Kannengießer N, Sturm B, Komarov MM, Stork W, Sunyaev A (2019) An open multimodal mobility platform based on distributed ledger technology. In: Internet of things, smart spaces, and next generation networks and systems: 19th international conference, NEW2AN 2019, and 12th conference, ruSMART 2019, St. Petersburg, Russia, August 26–28, 2019, Proceedings 19. Springer, pp 41–52

Lamport L (2001) Paxos made simple. ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001), 51–58
Lamport L, Shostak R, Pease M (2019) The byzantine generals problem. In: Concurrency: the works of Leslie Lamport, pp 203–226
Le Brun MA, Attard DP, Francalanza A (2021) Graft: general purpose raft consensus in elixir. In: Proceedings of the 20th ACM SIGPLAN international workshop on Erlang, pp 2–14
Li J, Wu J, Chen L (2018) Block-secure: blockchain based scheme for secure P2P cloud storage. Inf Sci 465:219–231
Liskov B, Cowling J (2012) Viewstamped replication revisited
Liu H, Luo X, Liu H, Xia X (2021) Merkle tree: A fundamental component of blockchains. In: 2021 international conference on electronic information engineering and computer science (EIECS). IEEE, pp 556–561
Menon AA, Saranya T, Sureshbabu S, Mahesh A (2022) A comparatıve analysis on three consensus algorithms: proof of burn, proof of elapsed time, proof of authority. In: Computer networks and inventive communication technologies: proceedings of fourth ICCNCT 2021. Springer, pp 369–383
Merkle RC (1988) A digital signature based on a conventional encryption function. In: Advances in cryptology—CRYPTO'87: proceedings. Springer, vol 7, pp 369–378
Mohamed AA, Ibrahim AO (2020) Blockchain consensuses algorithms based on proof of work: a comparative analysis. Int J Comput Commun Netw 2(1):12–20
Morais R, Crocker P, de Sousa SM (2020) A tool for implementing privacy in nano. In: 2020 IEEE international conference on decentralized applications and infrastructures (DAPPS). IEEE, pp 159–163
Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Decentralized business review, 21260
Nguyen CT, Hoang DT, Nguyen DN, Niyato D, Nguyen HT, Dutkiewicz E (2019) Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. IEEE Access 7:85727–85745
Oki BM, Liskov BH (1988) Viewstamped replication: a new primary copy method to support highly-available distributed systems. In: Proceedings of the seventh annual ACM symposium on principles of distributed computing, pp 8–17
Osadchuk M, Oliynykov R (2019) Method of proof of work consensus algorithms comparison. Radiotekhnika 198:105–112
Park S, Kwon A, Fuchsbauer G, Gaži P, Alwen J, Pietrzak K (2018) Spacemint: a cryptocurrency based on proofs of space. In: Financial cryptography and data security: 22nd international conference, FC 2018, Nieuwpoort, Curaçao, February 26–March 2, 2018, Revised Selected Papers 22. Springer, pp 480–499
Perard D, Lacan J, Bachy Y, Detchart J (2018) Erasure code-based low storage blockchain node. In: 2018 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). IEEE, pp 1622–1627
Puthal D, Mohanty SP (2018) Proof of authentication: IoT-friendly blockchains. IEEE Potentials 38(1):26–29
Puthal D, Malik N, Mohanty SP, Kougianos E, Das G (2018) Everything you wanted to know about the blockchain: its promise, components, processes, and problems. IEEE Consum Electron Mag 7(4):6–14
Rahman MU (2020) Scalable role-based access control using the EOS blockchain. arXiv preprint arXiv:2007.02163
Reddy S, Sharma G (2020) Ul-blockDAG: unsupervised learning based consensus protocol for blockchain. In: 2020 IEEE 40th international conference on distributed computing systems (ICDCS). IEEE, pp 1243–1248
Saad SMS, Radzi RZRM (2020) Comparative review of the blockchain consensus algorithm between proof of stake (POS) and delegated proof of stake (dPOS). Int J Innov Comput **10**(2)
Saad M, Qin Z, Ren K, Nyang D, Mohaisen D (2021) e-PoS: making proof-of-stake decentralized and fair. IEEE Trans Parallel Distrib Syst 32(8):1961–1973
Saez Y (2019) Coin.Ai: a proof-of-useful-work scheme for blockchain-based distributed deep learning. Entropy 21(8):723
Salimitari M, Chatterjee M, Fallah YP (2020) A survey on consensus methods in blockchain for resource-constrained IoT networks. Internet Things 11:100212
Sayeed S, Marco-Gisbert H (2019) Assessing blockchain consensus and security mechanisms against the 51% attack. Appl Sci 9(9):1788
Sayeed S, Marco-Gisbert H (2020) Proof of adjourn (PoAj): a novel approach to mitigate blockchain attacks. Appl Sci 10(18):6607
Sheth H, Dattani J (2019) Overview of blockchain technology. Asian J Convergence Technol AJCT. ISSN 2350-1146
Shrimali B, Patel HB (2022) Blockchain state-of-the-art: architecture, use cases, consensus, challenges and opportunities. J King Saud Univ Comput Inf Sci 34(9):6793–6807
Silvano WF, Marcelino R (2020) Iota tangle: a cryptocurrency to communicate internet-of-things data. Futur Gener Comput Syst 112:307–319
Singhal B, Dhameja G, Panda PS, Singhal B, Dhameja G, Panda PS (2018) How blockchain works. Beginning blockchain: a beginner's guide to building blockchain solutions, pp 31–148
Sunny J, Undralla N, Pillai VM (2020) Supply chain transparency through blockchain-based traceability: an overview with demonstration. Comput Ind Eng 150:106895
Tian S, Liu Y, Zhang Y, Zhao Y (2021) A byzantine fault-tolerant raft algorithm combined with Schnorr signature. In: 2021 15th international conference on ubiquitous information management and communication (IMCOM). IEEE, pp 1–5
Vashchuk O, Shuwar R (2018) Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake. Electron Inf Technol 9(9):106–112
Velliangiri S, Karthikeyan P (2020) Blockchain technology: challenges and security issues in consensus algorithm. In: 2020 International conference on computer communication and informatics (ICCCI). IEEE, pp 1–8
Vilim M, Duwe H, Kumar R (2016) Approximate bitcoin mining. In: 2016 53nd ACM/EDAC/IEEE design automation conference (DAC). IEEE, pp 1–6
Wang S, Tang X, Zhang Y, Chen J (2019) Auditable protocols for fair payment and physical asset delivery based on smart contracts. IEEE Access 7:109439–109453
Wang H, Guo K (2019) Byzantine fault tolerant algorithm based on vote. In: 2019 international conference on cyber-enabled distributed computing and knowledge discovery (CyberC). IEEE, pp 190–196
Wang D, Jin C, Li H, Perkowski M (2020) Proof of activity consensus algorithm based on credit reward mechanism. In: Web information systems and applications: 17th international conference, WISA 2020, Guangzhou, China, September 23–25, 2020, Proceedings. Springer, vol 17, pp 618–628
Wright A, De Filippi P (2015) Decentralized blockchain technology and the rise of lex cryptographia. SSRN 2580664
Wu Y, Song P, Wang F (2020) Hybrid consensus algorithm optimization: a mathematical method based on POS and PBFT and its application in blockchain. Math Probl Eng 2020
Xiang F, Huaimin W, Peichang S, Xue O, Xunhui Z (2021) Jointgraph: a DAG-based efficient consensus algorithm for consortium blockchains. Softw Pract Exp 51(10):1987–1999
Xiao Y, Zhang N, Li J, Lou W, Hou YT (2019) Distributed consensus protocols and algorithms. Blockchain Distrib Syst Secur 25:40
Xiong H, Chen M, Wu C, Zhao Y, Yi W (2022) Research on progress of blockchain consensus algorithm: a review on recent progress of blockchain consensus algorithms. Future Internet 14(2):47
Xu G, Liu Y, Khan PW (2019) Improvement of the DPoS consensus mechanism in blockchain based on vague sets. IEEE Trans Ind Inf 16(6):4252–4259
Xu M, Chen X, Kou G (2019) A systematic review of blockchain. Financ Innov 5(1):1–14
Yang F, Zhou W, Wu Q, Long R, Xiong NN, Zhou M (2019) Delegated proof of stake with downgrade: a secure and efficient blockchain consensus algorithm with downgrade mechanism. IEEE Access 7:118541–118555
Yang R, Wakefield R, Lyu S, Jayasuriya S, Han F, Yi X, Yang X, Amarasinghe G, Chen S (2020) Public and private blockchain in construction business process and information integration. Autom Constr 118:103276
Yang W, Garg S, Huang Z, Kang B (2022) A hybrid consensus algorithm for master-slave blockchain in a multidomain conversation system. Expert Syst Appl 204:117300
Yang X, Chen Y, Chen X (2019) Effective scheme against 51% attack on proof-of-work blockchain with history weighted information. In: 2019 IEEE international conference on blockchain (blockchain). IEEE, pp 261–265

Yusoff J, Mohamad Z, Anuar M (2022) A review: consensus algorithms on blockchain. J Comput Commun 10(9):37–50
Zeng L, Xin S, Xu A, Pang T, Yang T, Zheng M (2019) Seele's new anti-Asic consensus algorithm with emphasis on matrix computation. arXiv preprint arXiv:1905.04565
Zhang R, Xue R, Liu L (2019) Security and privacy on blockchain. ACM Comput Surv CSUR 52(3):1–34
Zheng X, Feng W (2021) Research on practical byzantine fault tolerant consensus algorithm based on blockchain. J Phys Conf Ser 1802:032022
Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE international congress on big data (BigData congress). IEEE, pp 557–564

## Publisher's Note