Hindawi
Security and Communication Networks
Volume 2022, Article ID 2812526, 13 pages
https://doi.org/10.1155/2022/2812526

Research Article

Survey of Consensus Algorithms for Proof of Stake in Blockchain

Lina Ge,[1,2,3] **Jie Wang**,[1,2] **and Guifen Zhang**[1]

[1]*School of Artificial Intelligence, Guangxi Minzu University, Nanning 530006, China*
[2]*Key Laboratory of Network Communication Engineering, Guangxi Minzu University, Nanning 530006, China*
[3]*Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning 530006, China*

Correspondence should be addressed to Lina Ge; 66436539@qq.com

Received 30 June 2021; Revised 22 April 2022; Accepted 10 May 2022; Published 29 May 2022

Academic Editor: Mohammad Ayoub Khan

As the core of blockchain technology, the consensus algorithm directly affects the security, stability, and decentralisation of the blockchain and numerous other important characteristics. Choosing an appropriate consensus algorithm for different scenarios is currently a challenge in the implementation of blockchain applications. This paper classifies the improvement schemes of proof of stake (PoS) into three categories: PoS-based consensus algorithms, PoS- and PoW-based consensus algorithms, and PoS- and BFT-based consensus algorithms. First, the study introduces the PoS and PoS consensus algorithm variants and then summarises the core ideas, effects, advantages, and disadvantages of these algorithms. Subsequently, the performances of the improved algorithms are compared. Finally, the main improved methods are summarised, and the most common network security attacks are discussed. The study lays a foundation for the main improvement directions of PoS in the future, hoping to provide a reference for researchers to help them select and design consensus algorithms in different application scenarios while also helping the evolution of consensus algorithms and the implementation of blockchain applications.

1. Introduction

Bitcoin [1] and Ethereum [2] achieve consensus among participants who do not trust each other, and blockchain has attracted the attention of the public [3–5]. Inspired by Bitcoin and cryptography, blockchain has emerged, evolved, and spread in several fields [6–10], such as finance [11], health [12], administration [13], industry [14], agriculture [15], smart cities [16, 17], and Internet-of-Things networks [18, 19]. Blockchain is a new type of technology that is integrated with a variety of computer technologies, such as distributed storage, peer-to-peer (P2P) networking, consistency verification, consensus algorithms, and cryptography. In this technology, blockchain data structures are used to verify and store data; consensus algorithms generate and update data; cryptography ensures the security of data transmission and access; and smart contracts composed of automated script codes program and manipulate data to realise trusted data management in an incompletely trusted environment. The consensus algorithm is the core of the blockchain and directly affects the efficiency, security, and stability of the entire system. Therefore, it is necessary to study consensus algorithms if blockchain technology is to become more widely used in the future.

The consensus algorithm ensures the safe and stable operation and consistency of the system. As the core of blockchain technology, the consensus algorithm stipulates the process of nodes keeping accounts by competition, that is, generates new blocks and then obtains a transaction fee [20].

The proof of stake (PoS) [21] was first proposed in July 2011 by a digital currency enthusiast at the Bitcoin Forum. However, the first implementation of PoS was in Peercoin (PPC), released by Sunny King in August 2012. PPC combines the two consensus algorithms of proof of work (PoW) and PoS. In the initial stage, PoW mining was used to distribute tokens relatively fairly to miners. Later, as the difficulty of mining increased, the system was mainly maintained by the PoS consensus algorithm. To a certain extent, PoS solves the problem of wasted power in PoW count and shortens the time to reach a consensus; therefore, following Bitcoin, many competing coins have adopted the

PoS consensus algorithm [22]. PoS obtains the bookkeeping right from the node with the highest equity, rather than from the node with the highest computing power in the system. Equity is reflected in the ownership of a specific amount of currency by a node, which is called currency age.

Unlike the computational power of PoW, PoS is realised by comparing the "coin age," which is determined by the amount of currency held by the node and the time of depositing the currency. PoS reduces the difficulty of computer hash calculations according to the size relation of the coin age and weight, alleviating resource waste to a certain extent.

2. Consensus Classification

2.1. Consensus Algorithm Based on PoS. To solve the problems of long-range attacks and nothing-at-stake attacks caused by forks, researchers have successfully developed improved algorithms based on PoS, such as Ouroboros, Sleepy Consensus, Snow White, and Delegated Proof of Stake (DPoS) [23]. The improved features in these algorithms are classified into three categories: (i) consensus algorithms to improve the enthusiasm for voting, (ii) consensus algorithms to guarantee the security of consensus in dynamic scenarios, and (iii) consensus algorithms to improve the efficiency of verification.

2.1.1. Improved Voting Motivation. Kiayias et al. proposed the Ouroboros algorithm in 2017. To encourage stakeholders to stay online and perform transaction verification and block production, they used a new reward mechanism to encourage nodes to join the blockchain to drive the PoS consensus process, making the behaviour of honest nodes in the chain approximate the Nash equilibrium, thereby effectively preventing block interception, selfish mining, and other attacks [21]. The existence of rational nodes is fully considered in the design of the incentive mechanisms. The transaction fees of multiple blocks are input into the pool and allocated to the corresponding nodes according to the contribution of the participating nodes [24]. This was the first PoS-based blockchain protocol with strict security guarantees.

To address the problems of inactive voting in DPoS and insufficient timely processing of malicious nodes, the authors in [25–27] proposed an improvement plan based on an incentive mechanism. In [25], two core schemes were proposed: a voting incentive mechanism and a checkpoint protocol. Through the voting incentive mechanism, token holders are encouraged to vote actively, and 101 relatively fair nodes are selected. In cooperation with the checkpoint protocol, malicious nodes are deleted in a timely manner by mutual inspection among the 101 nodes, thus improving efficiency. The introduction of a voting reward incentivises the nodes to vote, and the introduction of a reporting reward significantly increases the proportion of nodes that take the initiative to report. In addition, the resistance of ordinary nodes to the bribery of malicious nodes is enhanced, and the probability of malicious nodes becoming "agent nodes" is reduced, thus guaranteeing network security. Fu and Li [27] proposed an improvement scheme based on reward and credit mechanisms. In response to the low enthusiasm of nodes to vote in DPoS and to reduce the probability of malicious nodes being elected as representative nodes, the incentive mechanism suggests that the transaction fees earned by nodes be shared with their supporters, and the strategy of the node Shapley value plus the time factor be accordingly designed to redistribute the revenue. The combination of credit mechanism and punishment makes it more difficult for malicious nodes to become representative nodes.

2.1.2. Improved Dynamic Usability. In 2017, researchers at Cornell University developed a new algorithm called Sleepy Consensus [28]. This algorithm proves that the traditional consensus algorithm cannot guarantee the security of the consensus in a dynamic scenario (when a large number of nodes are offline). However, in an actual case, only a few nodes are online and participate in the consensus process. Thus, the Sleepy Consensus algorithm requires only the number of honest nodes online to exceed the number of failed nodes to guarantee security and robustness [21]. In the same year, David et al. presented the "Ouroboros Praos" for the first time, which provided security against fully adaptive corruption in a semisynchronous setting. The incentive system of Ouroboros Praos is the same as that of Ouroboros, but it improves the election method of block producers. The public verifiable identity of block producers in Ouroboros is improved to match that of block producers identified in private. Other nodes cannot judge the identity of block producers in this round before the block producers successfully generate new blocks, which effectively prevents block producers from possible bribery attacks or distributed denial-of-service (DDoS) attacks. In 2018, Badertscher et al. proposed a PoS blockchain protocol combinable with dynamic availability, called Ouroboros Genesis [29], which is designed to bootstrap the process when a new node joins the network and solves long-term attacks on PoS. Genesis retains the part of Ouroboros Praos [30] that uses a verifiable random function (VRF) to randomly select block producers and modifies according to the longest chain principle. It allows parties to safely join (or rejoin) the protocol execution using only the Genesis block information and captures the setting of dynamic availability. All parties are allowed to join or leave the system at will and can stay offline for a long time. In 2019, Kerber et al. proposed the first formal analysis of a blockchain protocol based on PoS privacy protection called Ouroboros Crypsinous [31]. This protocol ensures the consistency and activity of privacy, such that privacy is independent of any other protocol running during classification implementation. It can resist adaptive attacks owing to the subtlety of its design. In 2019, Daian et al. presented Snow White [32]. Snow White proposed a reconfigurable consensus algorithm suitable for PoS, whereby the nodes can join and exit the network randomly. The reconfiguration interval is short and prevents adversaries from later posterior corruption attacks.

2.1.3. Improved Validation Efficiency. In August 2013, the Bitshares [23] project proposed DPoS. The idea of this design is similar to the "board of directors' decision making." The

share rights held by each node in the system are equivalent to a single ballot. The ballot holders vote for their trusted representatives, and a "board of directors" is formed according to the voting results and the willingness of the nodes. The "board of directors" takes turns to package and settle transactions and sign (produce) new blocks. Compared to PoS, DPoS provides higher security in a short time and can verify transactions in seconds. If the PoW and PoS consensus methods are the accounting methods of "power competition" and "equity competition," respectively, then DPoS can be called the accounting method of "democratic centralism." It not only can solve the problem of PoW waste energy and joint mining, which poses a threat to the decentralisation of the system, but also can compensate for the disadvantage caused by participants, with bookkeeping interests in PoS, who may not want to participate in bookkeeping [22]. EOS [33] proposed a consensus algorithm based on the Byzantine fault-tolerant algorithm + DPoS (BFT-DPoS) in the "EOSIO Technical White Paper" released on 16 March 2018. EOS uses the BFT-DPoS protocol to make the block interval reach 0.5 s, which greatly shortens the time delay of cross-link communication and greatly increases the number of confirmed transactions per unit time. If such a mechanism is successfully implemented in future versions of EOS, it will undoubtedly be a solid step toward supporting innumerable users of blockchain technology.

The results of the comparative analysis based on the PoS consensus algorithm are listed in Table 1.

2.2. Hybrid Consensus Algorithm Based on PoS and PoW. In the consensus algorithm based on PoS and PoW, the problems solved by the related improved algorithms primarily include forks, nothing-at-stake attacks, and "the rich get richer" in PoS. The solutions to the corresponding problems mainly include introducing reward and punishment mechanisms and setting an upper limit. The reward and punishment mechanism generally involves a node paying a certain deposit before reaching consensus. If the node is found to be malicious, the deposit is confiscated. The setting of an upper limit aims to limit the coin age to prevent the right of a node in PoS from becoming too large, which can lead to that node becoming the "dominant one." The following section introduces the reward and punishment mechanisms and the setting of the upper limit of the consensus algorithm.

2.2.1. Incentive and Punishment Mechanism. Casper is a protocol used by Ethereum during the serenity phase. It was introduced in 2015 as an improved PoS mechanism and security-deposit-based economic consensus protocol. There are currently two versions of Casper: Casper the Friendly Ghost (CTFG), proposed by Vlad Zamfir in 2015 [34], and Casper Friendly Finality Gadget (CFFG) [35], proposed by Vitalik Buterin and Virgial Griffith in 2015. The former is a clear PoS consensus, whereas the latter is an organic combination of PoW and PoS consensus. CTFG is a chain-based PoS design, whereas CFFG combines a chain-based PoS design and a Byzantine fault-tolerant PoS [24]. Casper offers appropriate tools and regulations to readjust participant incentives [36, 37]. A salient feature of Casper is that each node must pay a certain amount of deposit before participating in block generation and consensus. In this way, malicious nodes risk forfeiting their deposits, thus damaging economic interests. Casper resists nothing-at-stake attacks but reduces the enthusiasm of many nodes to participate in block verification [38]. Casper the Friendly Ghost (CTFG), proposed by Vlad Zamjir in 2015, is one of the versions of Casper. It is a chain-based PoS design and is also based on the security and identity verification of the deposit. The issue of "nothing-at-stake" is addressed by the "betting" mechanism of the margin. In the case of incentive agreements, CTFG treats the consensus process as a cooperative game, ensuring that each node maximises its benefits in an alliance composed of 100% consensus nodes against attacks by most coalitions. In 2015, Buterin and Griffith proposed that Casper Friendly Finality Gadget (CFFG) [35] is a consensus algorithm that combines PoW and PoS. Similar to Casper, in CFFG, each node must pay a deposit to become a verifier, and each checkpoint must go through two rounds of validation before the final validation is completed. Each round of Casper needs to obtain more than one-third of the coin age verification of the entire network to determine the final result, and the CFFG needs to obtain more than two-thirds of the validators' legal votes. The main purpose of the betting mechanism in CFFG is to solve the problem of "nothing-at-stake" that the PoS consensus may face. To keep the nodes fully online, Ethereum adopted a penalty mechanism for offline nodes to maintain network security [24].

Proof of activity (PoA) was presented by Bentov et al. in 2014. The PoA protocol combines PoW and PoS and is an extension of the Bitcoin protocol. In the PoA protocol, the transaction rewards obtained by the stakeholders who generate the block are shared with the rest of the stakeholders and miners who generate empty block headers. Online miners can obtain a profit even if they do not mine, which encourages them to stay online and is conducive to the healthy operation of the currency. However, no specific scheme is provided for the income distribution of each node [39]. In 2016, PoA designers proposed the chain of activity (CoA) [40] protocol, which uses the idea of PoA to improve the PoS mechanism and overcome the bifurcation problem of PoS to a certain extent. The CoA execution process is similar to an online lottery in which all stakeholders draw prizes according to the CoA protocol. The distribution of rewards is the same as that of the original algorithm, and the benefits are shared by others. However, the specific distribution scheme of each person's benefits is not clear.

Duong et al. [41] proposed the 2-hop consensus algorithm in 2017. This is the first study to use the power of virtual resources to construct provably secure open blockchains. This is also the first attempt to combine physical and virtual resources to build a practical open blockchain with provable security. In the 2-hop protocol,

Table 1: Comparison of the consensus algorithms based on PoS.

Consensus	Main idea	Solution effect	Existing problems
DPoS	Witnesses take turns to generate blocks in sequence, and other witnesses verify the blocks	Provides higher security, can verify transactions in seconds, and can resist 51% attacks	Ordinary nodes do not actively participate in voting; corruption and bribery occur
Snow White	In each reconfiguration process, the nearest interest owner in the system is selected as the active member set, and the blocker is randomly selected according to the equity ratio	Meets the needs of nodes to join and exit the network randomly to achieve fairness	The network model is a sleepy model, and nodes cannot permanently remain online
Sleepy Consensus	The distributed protocol is studied in the "sleep" computing model; the online status of each node can be changed at any time	Can guarantee security and robustness when the number of active honest nodes reaches more than half or the number of honest online nodes exceeds the number of failed nodes	
Ouroboros	The reward mechanism encourages nodes to join the blockchain; a random function randomly selects block producers for each round of a period from the set of all current legal block producers	Randomly selects bookkeepers according to the stake, solving the long-range attacks	May be subject to bribery attacks or DDoS attacks
Ouroboros Praos	The block producer is determined privately, and other nodes cannot determine the block producer in this round before successfully generating a new block	Effectively prevents block producers from being attacked by bribery or DDoS attacks	
Ouroboros Genesis	When a new node joins the network, the selected blockchain needs to have a common prefix with the other chains and be the longest chain	Addresses long-range attacks, allowing parties to join and leave the system at will	
Ouroboros Crypsinous	The SNARK mechanism of "transaction injection" in Zerocash is extended to an environment where the currency does not depreciate so that no additional information is disclosed in the process	Implements for the first time analogue-based privacy that is universally composable and secure in the forward direction. Ensures consistency and mobility and is resistant to adaptive attacks	
EOS	Ballot holders elect the representatives they support by voting, and the witness network composed of these representatives reaches consensus through BFT	Makes the block generation interval reach 0.5 s and shortens the delay to 1.5 s, reaching millions of transactions per second	The actual throughput and decentralisation are not ideal
Improvement of DPoS	The voting incentive mechanism and the checkpoint protocol of PBFT are introduced to enhance community activity; malicious nodes are removed and punished in a timely manner	Accelerates the processing speed of malicious nodes, boosts operation efficiency, and generates block quickly	
Improvement of DPoS	Voting is used to reward incentive nodes actively participating in the voting, and reporting is used to reward incentive nodes actively reporting bribery nodes	Improves the enthusiasm of nodes to vote and strengthens the resistance of ordinary nodes to the bribery of malicious nodes	
Improvement of DPoS	The transaction fee of a node is shared with its voters, and it is difficult for a malicious node to become a representative node as the calculation of the voting results is optimised	Reduces the probability of the successful election of malicious nodes and improves the security of the system	Throughput and consensus delay are not ideal

the authors proposed a rigorous framework suitable for analysing more blockchain protocols. The 2-hop design can be viewed as a natural extension of Nakamoto's 1-hop design via a PoW mechanism (i.e., the second hop is deterministic and always true). The 2-hop design can also be viewed as a PoS scheme that uses a PoW chain as the biased random beacon.

2.2.2. Setting of an Upper Limit. In April 2014, Larry Ren proposed a consensus algorithm for proof-of-stake velocity (PoSV) in a white paper on ReddCoin, aiming to address the problem that coin age is a linear function of time in PoS, to eliminate the phenomenon of currency holders hoarding coins. In the first stage of the PoSV algorithm, PoW is used to realise token allocation, whereas in the second stage, PoSV

is used to maintain long-term network security. PoSV modifies the linear function of the coin age and time in PoS into an exponential decay function, and the growth rate of the coin age decreases with time and finally approaches zero. Therefore, the coin age of the new currency grows faster than that of the old currency until it reaches the upper threshold, which mitigates the phenomenon of currency hoarding by coin holders to a certain extent [22]. PoSV was proposed as an alternative to PoW and PoS to improve the security of P2P networks and confirm ReddCoin transactions. Proof of burn (PoB) was proposed in 2014 as a distributed consensus approach in which one cryptocurrency can be burned to create another. The subtlety of this version is that simulation of the mining platform and dependence on external randomness at low bit rates are not necessary. The PoB competes for the right to produce new blocks by burning tokens. In PoB, over time, the share of nodes in the system may decrease, driving nodes to burn tokens to obtain more mining opportunities. However, this causes a waste of token resources, and the mining ability is gradually controlled by those who have more resources and are willing to burn tokens [42]. Slimcoin [20] is an alternative cryptocurrency based on Peercoin, which uses PoB as part of the consistency algorithm as well as an alternative mining approach. In addition, Stewart's version of PoB is an attempted protocol that can be used in cryptocurrency to continuously generate blockchain or to mine [43].

Wu et al. proposed an improved blockchain consensus algorithm called proof of work and stake (PoWaS) in January 2020. PoWaS reduces the difficulty of hash calculation and sets the maximum difficulty value, the effective holding time, and the upper limit of the coin age, adjusting the credit value according to the behaviour of nodes, to finally introduce a competitive waiting time. The value of pStake is calculated according to the time, coin age, and credit value spent looking for random numbers. The calculation of pStake strengthens the impact of credit value on the competition to obtain accounting rights by reducing the proportion of equity and calculation power while increasing the proportion of credit value. PoWaS can reduce computing power waste, accelerate block output speed, and balance the competition for accounting rights [44]; however, there is still room for improvement in the stability and availability of block output speed.

The results of the comparative analysis of the hybrid consensus algorithm based on PoS and PoW are presented in Table 2.

2.3. Hybrid Consensus Algorithm Based on PoS and BFT. In the consensus algorithm based on PoS and BFT, the related improved algorithms include the strategy of combining PoS and BFT and the scheme of integrating a VRF on this basis, as analysed in this section.

2.3.1. Consensus Algorithm Based on PoS and BFT. Delegated Byzantine fault tolerance (dBFT) is a consensus algorithm adopted by NEO [45], which combines the delegated voting system, PoS, and practical BFT (PBFT) and is a consensus protocol for realising large-scale participation by delegated voting. This is equivalent to the people's congress system, and the bookkeeper is equivalent to the deputies of the people's congress. The deputies of the people's congress negotiate and decide on government affairs by the people's congress. dBFT can confirm newly generated blocks immediately; therefore, it is characterised by speed and good scalability. The generation speed of new blocks can reach 15–20 s, and the measured throughput can reach 1000 transactions per second (TPS). However, in the NEO project, six of the seven current consensus nodes are controlled by the project party. Therefore, there are also disadvantages, such as the limited number of consensus nodes and the very high degree of centralisation [46]. Tendermint [47], proposed by Gilad in 2014, implemented the first PBFT-based PoS consensus algorithm using blocks, hash links, dynamic validator sets, and a circular leader election. Based on the counting of votes by a node, a weight is assigned to each vote, and the problem of nothing-at-stake is solved by paying a deposit. Inspired by this, the BA* protocol in Algorand [47, 48] and CFFG protocol in Ethereum were proposed successively [34]. Tendermint is a Byzantine fault-tolerant consensus algorithm that is robust against double-spending attacks and can withstand up to one-third of the saboteurs in a network. Tendermint was applied to the CITA project [46]. Owing to its complex consensus algorithm, Tendermint does not have a corresponding real-world trust model. In 2016, Miller et al. introduced improvements to Tendermint and presented an alternative, HoneyBadgerBFT, which is the first practical asynchronous BFT protocol to guarantee liveness without making any timing assumptions. Their solution is based on a novel atomic broadcast protocol that achieves optimal asymptotic efficiency. Miller et al. presented an implementation and experimental results to show that their system can achieve throughput of tens of thousands of TPS and can scale to over 100 nodes on a wide area network (WAN). Unlike the alternatives, HoneyBadgerBFT does not consider the underlying network. Miller et al. conducted BFT experiments over Tor without the need to tune any parameters.

2.3.2. Consensus Algorithm Based on PoS, BFT, and VRF. The ontology project combines PoS, BFT, and VRF with the proposed VBFT [49] (Byzantine consensus algorithm based on VRFs), which realises fast consensus in the network. Each block determines the output of VRF. The VRF determines the sequence of consensus nodes, assigns priority according to the node sequence, determines the block priority by node priority weighting, and finally votes for the block with the highest priority to solve fork problems. The block producer, verification node, and confirmation node, which can resist malicious attacks and have a high degree of decentralisation and security, are randomly selected in VBFT. In 2017, Gilad et al. combined PoS, BFT, and VRF to propose the Algorand [50] consensus algorithm, which realises the rapid consensus of synchronous networks. This mechanism is equivalent to a multicommittee system that includes a block-producing node committee and

Table 2: Comparison of the hybrid consensus algorithms based on PoS and PoW.

Consensus	Main idea	Solution effect	Existing problems
Casper	Allocates and controls the margin to drive verifier validation and consensus. In the event of an "invalid" action, the deposit is forfeited, and participation in the consensus is not possible	Solves the "nothing-at-stake" problem. Ensures that the benefits are maximised in the alliance composed of 100% consensus nodes to resist the attack of the majority alliance	
CTFG	Allocates and controls the margin to drive verifier validation and consensus. If an "invalid" action occurs, the deposit is forfeited	Solves the "nothing-at-stake" problem and resists the attack of the majority alliance	
CFFG	The probability of a node being selected is proportional to the weight of the deposit, and at least two-thirds of the validators' legal votes are obtained in each round	Makes the nodes fully online, maintaining the security of the Ethereum network	Many nodes participate in betting on the verification of the initiative to reduce, preferring not to act
2-hop	PoS rights are introduced on the basis of PoW computing power so that blockchain security is built on the basis that honest nodes occupy the majority of joint resources	Solves 51% attacks and greatly improves the security of the blockchain	
PoA	The principle of the longest main chain is used to suppress forks. Parts of the PoW tokens are distributed to active nodes in a lottery. The rights and interests of nodes are proportional to the probability of winning	Has security higher than that of PoW and PoS, which encourages miners to keep running online and is conducive to the healthy operation of the currency	The verification of nodes in the entire network has become complicated, and there is no specific profit distribution plan
CoA	Within a certain period of time, a stakeholder is randomly selected to create a new block, similar to an online lottery	Overcomes the fork problem of PoS to a certain extent	There is no clear plan for the distribution of specific benefits
PoB	By burning tokens to compete for the right to produce new blocks, PoB makes the probability of a node being selected proportional to the number of burned tokens	Reduces dependence on low bit rate external randomness	There is a waste of token resources; mining capacity is gradually controlled by those who have more resources and are willing to burn tokens
PoSV	PoW is used to realise token distribution and uses PoSV to maintain long-term network security. The linear function of coin age and time is modified into an exponential decay function	Eliminates the phenomenon of currency hoarding by currency holders	
PoWaS	The difficulty of hash calculation is reduced; the upper limits for coin age and effective holding time are set; and the package accounting rights are determined by the time spent looking for random numbers, coin age, and credit value	Increases the difficulty of double spending and forks and prevents replay attacks	The stability of block speed is not ideal, and the adjustment of credit value needs to be further studied

verification node committee. It randomly selects block and verification nodes, which can resist malicious attacks and effectively prevent the verification power from being concentrated in the hands of some users. Algorand has a high degree of decentralisation and can guarantee security and activity during synchronisation. Kokoris-Kogias et al. [51] combined VRF, PoS, BFT, and a lock mechanism to propose the OmniLedger consensus algorithm, which realises the atomicity of cross-shard transactions from the perspective of the unspent transaction output (UTXO), whereby each shard chain has and maintains its own UTXO. It ensures security and correctness by using a bias-resistant public-randomness protocol to choose large statistically representative shards that process transactions and by introducing an efficient cross-shard commit protocol that atomically handles transactions affecting multiple shards. OmniLedger uses PoS to select verification nodes and VRF to allocate these nodes to the shard chains. Each shard chain uses a BFT to reach an agreement and uses a lock mechanism to ensure the atomicity and correctness of cross-shard chain operations. There is no corresponding real-world trust model to OmniLedger because of the complexity

of its mechanism. Abraham et al. [52] combined the VRF, PoS, and notarisation systems to propose the Dfinity consensus algorithm. Dfinity uses the Bone-h–Lynn–Shacham (BLS) threshold signature to construct a VRF (called a beacon) and outputs a data stream that changes over time. Dfinity is equivalent to a notarisation system, and the proposal can be considered credible after being certified by any legal notary. Byzantine nodes cannot secretly establish and maintain authenticated chains; therefore, there are no security threats such as double-spending attacks, selfish mining attacks, long-range attacks, and nonhazardous attacks, but there are adaptive attacks.

The results of the comparative analysis of the hybrid consensus algorithm based on PoS and BFT are presented in Table 3.

3. Consensus Comparison

The second section identifies and analyses the improved algorithms based on PoS. These algorithms have different design emphasis, and their advantages and disadvantages are presented in this section through comparisons and analyses from the perspectives of Byzantine fault tolerance, block generation speed, and throughput.

3.1. Fault Tolerance. As a reference for algorithm security, fault tolerance refers to the tolerance value of the consensus algorithm for nodes that have non-Byzantine faults in the system (crash fault tolerance) and the tolerance value of nodes that have Byzantine faults (Byzantine fault tolerance). The Byzantine fault tolerance of the nodes in the algorithm was compared to analyse security, and the results follow the following order: PoS = DPoS = Sleepy Consensus = Ouroboros = Casper = 2-hop = PoA = PoB = PoSV = PoS = DPoS = 50% > EOS = dBFT = Tendermint = HoneyBadger = 33.33%. The Byzantine fault tolerance of each algorithm is illustrated in Figure 1.

3.2. Block Time. The time it takes for a transaction to be packaged into a block and recorded in the blockchain is a performance indicator of whether a block is efficient. In this section, the algorithms are compared regarding the speed of reaching consensus (i.e., block time) and analysed with respect to efficiency. The results of the comparison show the following order: EOS > DPoS > VBFT > Dfinity = dBFT > OmniLedger > PoS. Compared with the original PoS algorithm, the improved algorithm significantly improves block generation time. VBFT consensus nodes execute the BFT consensus with low resource consumption and faster block generation; however, the scalability of VBFT decreases with an increase in consensus nodes [46]. Dfinity uses VRF to generate proposers, which improves the security of the system; however, communication via broadcast is time complex. The block generation speeds for each algorithm are displayed in Table 4 and Figure 2.

3.3. Throughput. Transaction throughput is defined as the number of TPS in the blockchain, which is a key index for measuring the performance of a system. The scalability of the network is one of the key factors to consider in blockchain design and can often be determined from the throughput. The faster the block generation speed of the algorithm used in the actual system, the greater the transaction throughput and the higher the performance efficiency of the algorithm:

$$\mathrm{TPS}_{\Delta t} = \frac{T\_\mathrm{Transactions}_{\Delta t}}{\triangle t}, \tag{1}$$

where $T\_\mathrm{Transactions}_{\Delta t}$ represents the total transaction volume per unit time t and Δt represents the time interval between the creation of a transaction and the confirmation of the block. The results of the throughput comparison yield the following order: DPoS > OmniLedger > HoneyBadger = VBFT > PoS = dBFT = Dfinity > Algorithm > Snow White- > EOS. Compared to the throughput of the original algorithm, the throughputs of most of the improved algorithms were significantly enhanced. The block generation speed and throughput performance of the DPoS in the experimental environment were better than those of the other algorithms. The throughput of a few improved algorithms decreased in comparison with that of the original algorithm, whereas some improved algorithms maintained the throughput of the original algorithm. Snow White sacrifices high throughput so that nodes can randomly join and exit the network and to ensure fairness of the transaction fee distribution. Algorand sacrifices high throughput to ensure rapid consensus of the synchronisation network and a high degree of decentralisation. Although EOS theoretically reached million-level TPS, its actual throughput was not ideal because the process of electing witnesses consumed a large amount of the resources. The throughputs of the improved algorithms are presented in Table 5 and Figure 3.

4. Discussion and Challenges

In this section, the improvement methods of the consensus algorithms are discussed, followed by some of the challenges faced by blockchain.

4.1. Improvement Methods. Through a study of the PoS algorithm, it was determined that the improvement in the PoS algorithm is mainly obtained by focusing on how to select the block producers, distribute the block rewards, incentivise nodes to participate in the consensus, impose sanctions on lazy or malicious nodes, and prevent excessive concentration of power. The details of these processes are as follows.

4.1.1. Selection of the Block Producer. First, the election of blockmakers, which can be through direct or indirect election, is considered. Some protocols directly select a node from all the nodes as the producer of the new block, and some protocols select multiple nodes from all the nodes to form a node set. A node is selected from this set as the block producer.

Table 3: Comparison of the hybrid consensus algorithms based on PoS and BFT.

Consensus	Main idea	Solution effect	Existing problems
dBFT	Participants vote on nodes according to the tokens they hold and elect bookkeepers, and all bookkeepers run the BFT algorithm to reach a consensus to generate a new block	The newly generated block can be confirmed immediately, a new block is generated every 15–20 s, and throughput can reach 1000 TPS	The number of consensus nodes is limited, and the degree of centralisation is too high
Tendermint	On the basis of counting votes by a node, a weight is assigned to each vote, and the deposit is forfeited if a node behaves maliciously	It is robust against double-spending attacks and can resist up to one-third of the saboteurs in the network	There is no corresponding real-world trust model. It has high time complexity
HoneyBadger	The consensus is based on an atomic broadcast that can achieve asymptotic validity and can process tens of thousands of TPS on hundreds of nodes in the WAN	Extended by hundreds of nodes, achieving a throughput of tens of thousands of TPS	
VBFT	All legitimate voters have the right to vote and be voted, and the legitimacy of voter identity is verified by a VRF	Solves the bifurcation problem, realises the rapid consensus of the network, resists malicious attacks, and has a high degree of decentralisation and security	With the increase in consensus nodes, the scalability decreases
Algorand	VRF randomly elects block nodes and verification nodes and determines a verifier set and leader in the form of an encrypted lottery (the one with the smallest credential value is elected)	Resists malicious attacks, has low resource consumption and a high degree of decentralisation, and can ensure security and activity during synchronisation	It has higher calculation complexity and communication overhead than those of PBFT
OmniLedger	PoS selects verification nodes; VRF assigns verification nodes to the shard chain; and the shard chain reaches an agreement through BFT	Can resist Sybil attacks; the throughput increases linearly with the number of shard chains	
Dfinity	BLS generates random numbers, and a group of people generate signatures. A single person cannot prevent the issuance of the signature, and no individual can predict the result of the signature	Effectively prevents double-spending attack, selfish mining attack, long-range attack, and noninterest attack	There are adaptive attacks; it uses broadcast communication and has high time complexity

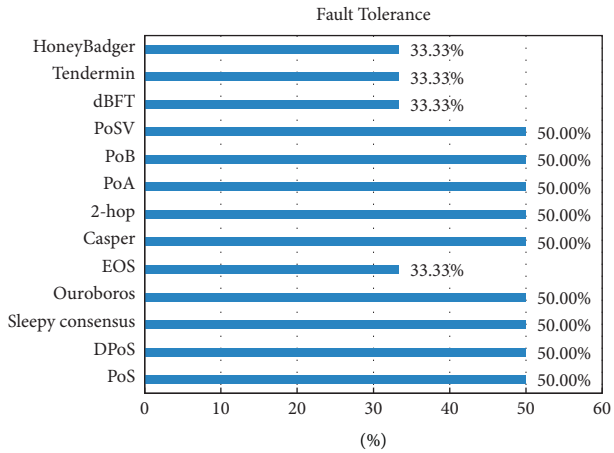

Figure 1: Byzantine fault tolerance.

Second, the election method of block producers is considered for two situations. The first is the situation in which block producers are elected directly from all nodes. The main election methods are capable of competition, and those with greater abilities are elected as block producers. For example, PoW is a competition for computing power; PoS is a competition for stakes; PoB is a competition for burning tokens; and PoWaS is a competition for pStake (pStake is calculated according to the time, coin age, and credit value spent looking for random numbers). The election of block producers through competition can easily lead to a dominant situation in the blockchain. For example,

Table 4: Block generation speeds.

Consensus	PoS [21]	DPoS [23]	EOS [33]	dBFT [45]	VBFT [49]	OmniLedger [51]	Dfinity [52]
Speed (s)	64	3	0.5	15	5–10	63	5–10

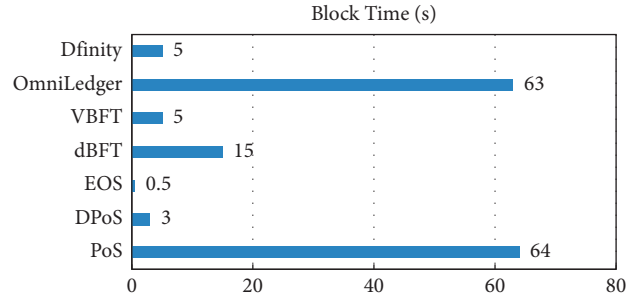

Figure 2: Comparison of block generation speeds.

Table 5: Throughput of consensus algorithm.

Consensus	PoS [21]	DPoS [23]	Snow White [32]	EOS [33]	dBFT [45]	HoneyBadger [53]	VBFT [49]	Algorand [50]	OmniLedger [51]	Dfinity [52]
Speed (s)	10^3	10^6	1.25×10^3	34	10^3	3×10^3	3×10^3	8.75×10^2	3.5×10^3	10^3

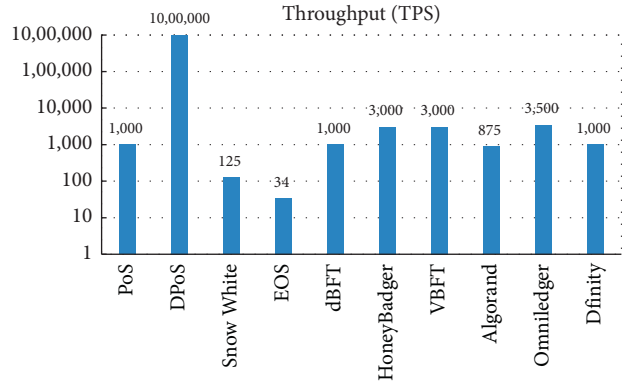

Figure 3: Comparison of throughput.

a node with a large stake in the PoS has a greater chance of producing a new block. After a block reward is received, the stake of the node becomes larger, which leads to the problem of "the rich get richer." The same problem also arises with PoW and PoB. The second situation is the indirect selection of blocks from a set of nodes. There are three ways to elect a block producer in the set: vote to elect the block producer, take turns as the block producer, and randomly designate a node as the block producer.

(i) The voting method to generate block producers in the set is similar to the committee mechanism. Decisions are made by committee members to vote for the resolution, and some innovative protocols assign different weights to the votes. For example, the EOS block sequence is determined based on the network resources of each node, and this order is valid only when the consent of at least 15 block producers is obtained. dBFT generates a new block by consensus by running the BFT algorithm through a collection of bookkeepers. VBPT is similar to the block-producing election of Dfinity.

(ii) The manner in which the nodes in the set take turns as block producers is also similar to the committee mechanism. For example, the collective "board of directors" in the DPoS takes turns to package and settle transactions and sign (produce) new blocks according to the established schedule.

(iii) The block producers are elected by VRF. For example, the consensus of the Ouroboros and Ouroboros series uses random numbers to select block producers randomly in each round. The CoA randomly selects a node from the set of stakeholders as a block producer, and Algorand randomly elects block producers and verification nodes. Electing block producers in a random manner can resist malicious attacks and provide a high degree of decentralisation and security.

This mode of election is relatively fair and can prevent the problem of one-party dominance to a certain extent; however, it also has some disadvantages. In theory, DPoS has excellent throughput; however, the throughput of EOS applications is not ideal, and the block generation relies on 21 witnesses; therefore, decentralisation is not good; dBFT is close to being completely centralised in the NEO project. The resource consumption of the VBPT is low; however, its scalability deteriorates with an increase in the number of nodes. Dfinity uses broadcasting to communicate during the consensus process, which leads to high time complexity while improving system security. Algorand performs well in

terms of decentralisation, security, and resource consumption but has high communication overhead with at least six rounds of communication.

4.1.2. Allocation of Block Rewards. Some protocols clarify how rewards should be distributed, whereas others do not specify the distribution method for block rewards (PoA and CoA only indicate that rewards are shared with n potential block producers, but specific plans are not provided).

There are two main types of reward-allocation schemes: First, the reward is exclusive to the block producer. In general, the ability competition agreement and reward are exclusive to block producers such as PoS, PoA, and PoB. Second, the reward is shared with the block producer and others, such as Ouroboros and Ouroboros Praos, who allocate transaction fees to participating nodes according to the amount they contribute, whereas Casper rewards according to the amount of money the verifiers place.

The exclusive allocation of a block reward leads to the concentration of power in the blockchain network. The equal allocation of a block reward is obviously unfair to nodes with different contribution levels, and the allocation of reward according to the "bet" also leads to concentration of power to some extent. Thus, the questions of allocating the reward more reasonably, that is, how to allocate the reward according to the contribution of each node and how to define the measurement index of contribution, are also key issues in the design of reward allocation in the consensus algorithm.

4.1.3. Incentive and Punishment Mechanism. Generating a new block in the blockchain requires agreement among the online nodes. The reward mechanism is primarily intended to encourage inert nodes (nodes that are not online or those that are not actively participating in the consensus) to actively participate in the consensus. The penalty mechanism is mainly intended to punish malicious nodes (nodes that violate consensus rules, compromise consensus security, or do not actively participate in consensus). The reward and punishment mechanisms aim to reach a consensus in a healthy manner. For example, to encourage nodes to stay online, Ouroboros places the transaction fees of multiple blocks into the pool and allocates transaction fees according to the contribution of the participating nodes. PoWaS rewards and penalises according to the credit value upgrade mechanism of the node to promote active competition for the right of bookkeeping by the node. The Casper series and Tendermint protocols impose sanctions on malicious nodes by paying a deposit, which can be forfeited through ownership. CoA penalises nodes for "inaction" through the "three times" blacklisting rule. If the nodes fail to produce blocks three consecutive times, they are blacklisted; however, they can be removed from the blacklist after resuming normal block production.

The original intention of designing the reward and punishment mechanism is to ensure the healthy operation of the consensus algorithm, motivate nodes to actively reach consensus, and punish malicious nodes to guarantee that the consensus algorithm operates under safe conditions. However, controlling the punishment strategy and scale is challenging. Malicious nodes should be punished while not allowing some inert nodes to go unpunished for not participating in the consensus, not mining, or even quitting the network. Therefore, the adaptation and optimisation of the reward and punishment mechanisms are also problems that researchers need to consider.

4.1.4. Setting an Upper Limit. The aim here is mainly to prevent the problem of excessive concentration of power, which could lead to "the rich getting richer" and other problems. PoSV uses an exponential decay function to decrease the growth rate of the coin age over time, which eventually tends toward zero. Setting an upper threshold for coin age alleviates the phenomenon of hoarding coins to a certain extent. PoWaS sets the upper limit for the effective holding time and coin age. If the upper limit is exceeded, the coin age and effective holding time will stop growing, thereby preventing the problems of unlimited growth of currency age and the "the rich getting richer." Setting an upper limit prevents the problem of excessive concentration of power but sacrifices block speed and stability in performance.

The above four aspects are the main improvement points summarised in this study of the PoS algorithm. The core problem is the selection of block producers and distribution of block rewards. The incentive and punishment mechanism and the setting of an upper limit somewhat answer the first two questions, that is, to elect block producers more reasonably and allocate block rewards more equitably.

In summary, the main improvement directions of the PoS-improved algorithm are outlined in Figure 4 for clarification.

4.2. Blockchain Consensus Attacks. In this section, the most common network security attacks that theoretically threaten almost all consensus algorithms [54] are discussed. When designing a consensus algorithm, more attention should be paid to possible attacks on different blockchain types.

4.2.1. Double-Spending Attack. A double-spending attack on the blockchain occurs when a node tries to spend a sum of money twice. The attacker first creates a normal transaction in the block of the main chain, then creates a deceptive transaction after a certain period of time, and publishes the deceptive transaction in the block of the fork chain to fork off from there. The attacker continues to mine on the Internet until the length of the fork chain exceeds the main chain. At this time, it broadcasts to the entire network, and once other nodes find out that there is a longer chain in the network, all of them switch to this fork chain, whereby the forked chain becomes the main chain; the previous normal transaction is rolled back, and the double-spending attack succeeds.

Although different consensus algorithms attempt to mitigate this vulnerability and have different mechanisms to

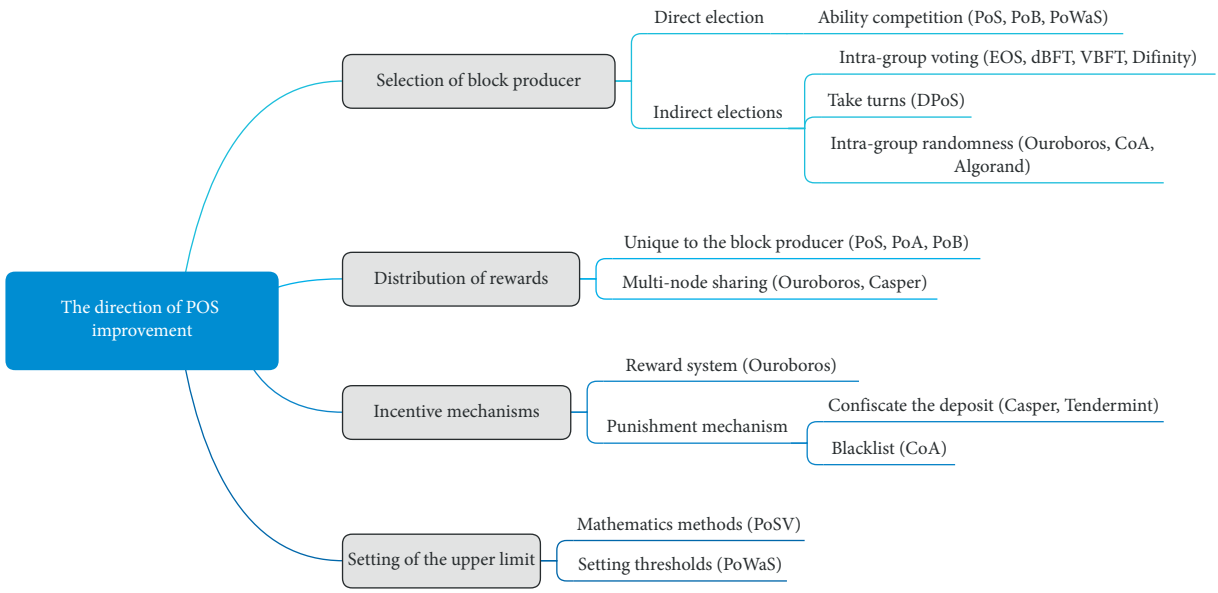

Figure 4: Main improvement directions of PoS.

address this issue, the double-spending attack cannot be completely avoided in blockchain systems; theoretically, it can happen at any time [55, 56]. To solve this problem, we can start by solving the time problem, preventing the attacker's malicious fork from becoming the longest main chain, and solving it through timestamp or nonce.

4.2.2. 51% Attack. From a theoretical perspective, 51% attacks are possible. When an attacker can control more than 50% of the power (such as mining power or verification power) in the blockchain, he/she can perform malicious activities [57]. Let us assume that a certain node has sufficiently strong computing power (more than half of the total computing power of the nodes in the entire network). If the node is mining on the forked chain, the growth rate of the forked chain will be greater than that of the main chain. At this point, if the node wants to roll back the transaction in the main chain, it only needs to publish the fork chain. Nodes with insufficient computing power can organise multiple nodes to launch 51% attacks through bribery.

In comparison, Sayeed and Marco-Gisbert [58] pointed out that the PoW, PoS, and DPoS algorithms are vulnerable to 51% attacks. In the PoA algorithm, attackers launching 51% attacks need to own more than half of the property and computing power in the blockchain network simultaneously, which increases the cost of attacks, preventing 51% attacks to a certain extent.

4.2.3. Sybil Attack. A Sybil attack is an attack that acts on a P2P network. In the blockchain, the attacker uses a single node to forge multiple identities to obtain voting rights and the ability to verify blocks or even broadcast a fake message to the blockchain network, thereby weakening the redundancy of the network and allowing the monitoring of normal activities of the network to interfere.

Although Sybil attacks are difficult to detect, some preventive methods have been developed. For example, PoB increases the cost of creating nodes by burning tokens to reduce the risk of attack. The PoW determines the voting rights of each user according to multiple parameters to defend against Sybil attacks [59].

4.2.4. Selfish Mining. Selfish mining is a strategy aimed at the Bitcoin PoW mechanism blockchain. Its purpose is not to destroy the operating mechanism of Bitcoin but to obtain additional rewards and make honest miners perform invalid calculations. Simply, when a block is discovered, the selfish miner does not announce but continues digging or alternatively waits for an opportunity before making the announcement or delays it deliberately. Therefore, it is possible to construct a private branch that selfish miners control, thus causing the chain to fork. This strategy reduces the speed of network verification of blocks while weakening the profitability of honest miners. Prior to difficulty adjustment, it also has an adverse effect on selfish miners. A selfish mining attack is an attack on the difficulty adjustment algorithm [60], which can be mitigated by introducing orphan blocks to the difficulty adjustment formula.

5. Conclusion

In the field of information technology, the consensus algorithm for blockchains has attracted increasing attention from researchers. This study examines PoS-improved algorithms and classifies them into three major categories. By introducing the basic concepts of the algorithm, the study summarises the improvement strategy, improvement effect, advantages, and disadvantages of four more algorithms, concentrating on improvement points and analysing and summarising them. The improvements in the algorithm are intended to solve the problem of electing block producers,

allocating block rewards reasonably, and improving consensus efficiency while ensuring privacy and security. The question is how to ensure that the blockchain remains decentralised while running the consensus without violating the previous conditions? Improvements to the algorithm should consider this point.

Blockchain technology has been applied in many fields, and an increasing number of researchers are focusing on the implementation of blockchain applications. Security and privacy issues in the process of application implementation remain major challenges. Based on the improved method in this paper, the plan is to design a consensus algorithm based on an incentive mechanism to distribute block rewards reasonably while preventing network security issues facing the blockchain.

Data Availability

Data are available upon request.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61862007) and the Innovation Project of Guangxi Minzu University Graduate Education (gxun-chxps202081).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2019, https://bitcoin.org/bitcoin.pdf.

[2] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.

[3] Y. Xu and Y. Huang, "MWPoW: multiple Winners Proof of Work Protocol, a decentralisation strengthened fast-confirm blockchain protocol," *Security and Communication Networks*, vol. 2019, Article ID 3674274, 13 pages, 2019.

[4] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[5] M. T. Quasim, M. A. Khan, F. Algarni, and A. Alharthy, G. M. M. Alshmrani, "Blockchain frameworks," in *Decentralised Internet of Things. Studies in Big Data*, M. Khan, M. Quasim, F. Algarni, and A. Alharthi, Eds., vol. 71, Berlin, Germany, Springer, 2020.

[6] M. A. Rashid, K. Deo, D. Prasad, K. Singh, S. Chand, and M. Assaf, "TEduChain: a platform for crowdsourcing tertiary education fund using blockchain technology," *International Journal of Production Research*, vol. 23, pp. 1–19, 2019.

[7] Z. Huang, X. X. Li, X. J. Lai, and K. F. Chen, "Blockchain and its application," *Journal of Information Security Research*, vol. 3, no. 3, pp. 237–245, 2017.

[8] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: using blockchain to protect personal data," in *Proceedings of the IEEE Security & Privacy Workshops*, pp. 180–184, San Jose, CA, USA, May 2015.

[9] J. Kang, R. Yu, X. Huang, and S. Y. E. Maharjan, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[10] Z. Meng, T. Morizumi, S. Miyata, and H. Kinoshita, "Design scheme of copyright management system based on digital watermarking and blockchain," in *Proceedings of the IEEE Computer Software & Applications Conference*, Tokyo, Japan, July 2018.

[11] A. Polyviou, P. Velanas, and J. Soldatos, "Blockchain technology: financial sector applications beyond cryptocurrencies," *The 3rd Annual Decentralized Conference on Blockchain and Cryptocurrency*, vol. 28, no. 1, p. 7, 2019.

[12] C. Agbo, Q. Mahmoud, and J. Eklund, "Blockchain technology in healthcare: a systematic review," *Healthcare*, vol. 7, no. 2, p. 56, 2019.

[13] V. Paliwal, S. Chandra, and S. Sharma, "Blockchain technology for sustainable supply chain management: a systematic literature review and a classification framework," *Sustainability*, vol. 12, no. 18, p. 7638, 2020.

[14] J. Lee, M. Azamfar, and J. Singh, "A blockchain enabled cyber-physical system architecture for Industry 4.0 manufacturing systems," *Manufacturing Letters*, vol. 20, pp. 34–39, 2019.

[15] G. Pau, M. Collotta, A. Ruano, and J. Qin, "Smart home energy management," *Energies*, vol. 10, no. 3, pp. 382–386, 2017.

[16] G. Mirabelli and V. Solina, "Blockchain and agricultural supply chains traceability: research trends and future challenges," *Procedia Manufacturing*, vol. 42, pp. 414–421, 2020.

[17] T. Frikha, F. Chaabane, N. Aouinti, O. Cheikhrouhou, N. B. Amor, and A. Kerrouche, "Implementation of blockchain consensus algorithm on embedded architecture," *Security and Communication Networks*, vol. 2021, Article ID 9918697, 11 pages, 2021.

[18] Y. Yu, Y. Li, J. Tian, and J. Liu, "Blockchain-based solutions to security and privacy issues in the internet of things," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 12–18, 2018.

[19] M. A. Khan, M. T. Quasim, F. Algarni, and A. Alharthi, *Decentralised Internet of Things: A Blockchain Perspective*, Berlin, Germany, 2020.

[20] Y. Liu, J. Ke, H. Jiang, and X. Song, "Improvement of the PoS consensus algorithm in blockchain based on Shapley value," *Journal of Computer Research and Development*, vol. 55, no. 10, pp. 2208–2218, 2018.

[21] E. W. Paper, "A next-generation smart contractand decentralized application platform [EB/OL]," 2020, https://github.com/ethereum/wiki/wiki/White-Paper.

[22] Y. Yuan, X.-C. Ni, S. Zeng, and F. Y. Wang, "Blockchain consensus algorithms: the state of the art and future trends," *Acta Automatica Sinica*, vol. 44, no. 11, pp. 2011–2022, 2018.

[23] D. Larimer, *Delegated Proof-Of-Stake (DPOS)*, https://www.geeksforgeeks.org/delegated-proof-of-stake/, 2019.

[24] Y. Z. Liu, J. W. Liu, Z. Y. Zhang, T. G. Xu, and H. Yu, "Overview on blockchain consensus algorithms," *Journal of Cryptologic Research*, vol. 6, no. 4, pp. 395–432, 2019.

[25] P. Tan Sen and C. Yang, "Research and improvement of blockchain DPoS consensus algorithm," *Modern Computer*, vol. 6, pp. 11–14, 2019.

[26] M. Chen, Y. Lin, and W. Lan, "Improvement of DPoS consensus algorithm based on 'reward system'," *Computer Science*, vol. 47, no. 2, pp. 269–275, 2020.

[27] Y. Fu and S. Li, "Improved plan for the consensus algorithm of authorized share certification," *Computer Engineering and Applications*, vol. 56, no. 19, pp. 48–54, 2020.

[28] R. Pass and E. Shi, "The sleepy model of consensus," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 1–55, Springer, Hong Kong, China, December 2017.

[29] C. Badertscher, P. Gai, A. Kiayias, Z. Vassilis, and A. Russell, "Ouroboros Genesis: composable proof-of-stake blockchains with dynamic availability," in *Proceedings of the 2018 ACM SIGSAC Conference*, pp. 1–66, New York, NY, USA, June 2018.

[30] B. David, P. Gaži, A. Kiayias, and A. Russell, "Ouroboros Praos: an adaptively-secure, semi-synchronous proof-of-stake blockchain," in *Proceedings of the International Conference on the Theory & Applications of Cryptographic Techniques*, pp. 1–37, Springer, Tel Aviv, Israel, May 2018.

[31] T. Kerber, M. Kohlweiss, A. Kiayias, M. Kohlweiss, and V. Zikas, "Ouroboros crypsinous: privacy-preserving proof-of-stake," in *Proceedings of the IACR Cryptology, ePrint Archive*, San Francisco, CA, USA, May 2018.

[32] P. Daian, R. Pass, E. Shi, and S. White, *Robustly Reconfigurable Consensus And Applications To Provably Secure Proofs Of Stake*IACR Cryptology ePrint Archive, 2017, https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md#consensus-algorithm-bft-dpos.

[33] EosIo, "Technical white paper v2," 2018, https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md#consensus-algorithm-bft-dpos.

[34] V. Zamfir, "Introducing casper 'the friendly Ghost'," 2015, https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/.

[35] V. Buterin and V. Griffith, "Casper the friendly finality Gadget," pp. 1–10, 2017, https://arxiv.org/abs/1710.09437.

[36] V. Buterin, D. Reijsbergen, S. Leonardos, and G. Piliouras, "Incentives in Ethereum's hybrid Casper protocol," in *Proceedings of the. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, pp. 236–244, Hoboken, NJ, USA, May 2019.

[37] K. Tsoulias, G. Palaiokrassas, G. Fragkos, A. Litke, and T. A. Varvarigou, "A graph model based blockchain implementation for increasing performance and security in decentralized ledger systems," *IEEE Access*, vol. 8, Article ID 130952, 2020.

[38] M. Zheng, H. Wang, H. Liu et al., "Survey on consensus algorithms of blockchain," *Netinfo Security*, vol. 19, no. 7, pp. 8–24, 2019.

[39] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity," *ACM SIGMETRICS - Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, 2014.

[40] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Proceedings of the International Conference on Financial Cryptography & Data Security*, February 2016.

[41] T. Duong, L. Fan, and H. S. Zhou, "2-hop blockchain: combining proof-of-work and proof-of-stake securely," pp. 1–45, 2017, https://eprint.iacr.org/2016/716.

[42] J. H. Huang, X. Xia, Z. C. Li, J. H. Li, and H. Zheng, "Proof of trust: mechanism of trust degree based on dynamic authorization," *Journal of Software*, vol. 30, no. 9, pp. 2593–2607, 2019, in Chinese.

[43] Bitcoin, "Proof of burn," 2018, https://en.bitcoin.it/wiki/Proof_of_burn.

[44] M. Wu, G. Zhu, and S. Wu, "Improved consensus algorithm of blockchain based on proof-of-work and proof-of-stake," *Journal of Computer Applications*, vol. 40, no. 8, pp. 2274–2278, 2020.

[45] Neo, "NEO white paper," 2014, https://docs.neo.org/zh-cn/white paper.html.

[46] M. Tan, J. Yang, L. Ding, X. Li, and S. Xia, "Summarization of blockchain consensus algorithm," *Computer Engineering*, vol. 46, no. 12, pp. 1–11, 2020.

[47] J. Kwon, "Tendermint: consensus without mining," 2014, https://diyhpl.us/~bryan/papers2/bitcoin/tendermint_v03.pdf.

[48] J. Chen and S. Micali, "Algorand: a secure and efficient distributed ledger," *Theoretical Computer Science*, vol. 777, pp. 155–183, 2019.

[49] A. Consensus, "Ontology project. Consensus algorithm," 2020, https://docs.ont.io/ontology-elements/consensus-mechanism.

[50] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: scaling Byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 51–68, IEEE Press, Washington DC, U.S.A, October 2017.

[51] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: a secure, scale-out, decentralized ledger via sharding," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy*, pp. 583–598, IEEE Press, San Francisco, CA, U.S.A, May 2018.

[52] I. Abraham, D. Malkhi, K. Nayak, and L. Ren, "Dfinity consensus, explored," 2018, https://eprint.iacr.org/2018/1153.pdf.

[53] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 31–42, ACM, New York, NY, U.S.A, October 2016.

[54] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2019.

[55] S. M. H. Bamakan, A. Motavali, and A. B. Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, Article ID 113385, 2020.

[56] H. Hasanova, U. J. Baek, M. G. Shin, K. Cho, and M. S. Kim, "A survey on blockchain cybersecurity vulnerabilities and possible countermeasures," *International Journal of Network Management*, vol. 29, no. 2, Article ID e2060, 2019.

[57] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[58] S. Sayeed and H. M. Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Applied Sciences*, vol. 9, no. 9, p. 1788, 2019.

[59] P. Compare, "What is proof of weight?," 2019, https://coincodex.com/article/2617/what-is-proof-of-weight/.

[60] C. Grunspan and R. P. Marco, "On profitability of selfish mining," 2018, https://arxiv.org/pdf/1805.08281.pdf.