IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

A Comprehensive Review of Blockchain Consensus Mechanisms

BAHAREH LASHKARI[1], PETR MUSILEK[1,2](Senior Member, IEEE)
[1]Department of Electrical and Computer Engineering, University of Alberta, Canada
[2]Department of Applied Cybernetics, University of Hradec Králové, Hradec Králové, Czech Republic

Corresponding author: Bahareh Lashkari (e-mail: bahareh1@ualberta.ca).

The support provided by the Government of Alberta under the Major Innovation Fund, project RCP-19-001-MIF, is gratefully acknowledged.

ABSTRACT Since the advent of distributed ledger technologies, they have provided diverse opportunities in a wide range of application domains. This article brings a comprehensive review of the fundamentals of distributed ledger and its variants. Analyzing 185 publications, ranging from academic journals to industry websites, it provides a comparative analysis of 130 consensus algorithms using a novel architectural classification. The distribution of the reviewed algorithms is analyzed in terms of the proposed classification and different application domains, along with the applicability of each class among the top 10 platforms in the most prominent blockchain application domains. Additional conclusions are drawn from the evolution of consensus mechanisms, and the analysis concludes envisaging future prospects for consensus as an important part of distributed ledger technology.

INDEX TERMS Blockchain, distributed ledger technology, consensus mechanisms, cryptocurrency

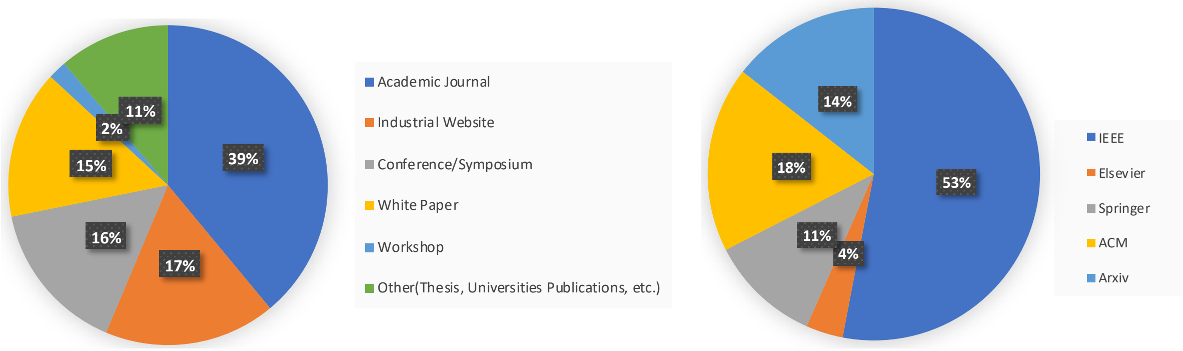
I. INTRODUCTION

THE emergence of ledgers can be tracked back more than thousands of years. It was followed by a conventional banking system where data records have been authenticated by a central authority. With the advent of computers, ledgers became digitized and evolved the preceding centralized ledger banking system, mirroring what was initially carried out on paper. A few years later, the distributed ledger technology has been proposed by Satoshi Nakamoto with the intention of excluding the former authoritative environments towards a verifiable structure. Distributed ledger technology (DLT) enabled a novel form of recording transactions using cryptography, advanced algorithms, and massive computing capacity. As a digital database instance, DLT is shared between individuals with certain characteristics that not only preserve particular communication protocols but also go through an agreement procedure that leads to a shared decision exclusive to the group of individuals that operate the DLT.

This paper reviews the fundamentals of DLT that led to the advent of blockchain and provides a comparative analysis of the 130 most recently introduced consensus mechanisms. We propose an architectural classification of consensus mechanisms that not only allows the examination of the existing consensuses but also provides a structure that subsequent algorithms can be related to. The current studies in this area have either analyzed a confined number of algorithms without any concerns regarding the features that are in common among consensus algorithms, or they have studied the algorithms in particular categories.

Previously, there have been several other attempts to analyse consensus mechanisms with the goal of establishing their taxonomy. Hattab [1] classifies the consensus mechanisms into 3 groups based on hardware, stake, and vote. In another study, performed by Chaudhry [2], the consensus mechanisms are categorized with respect to their scalability, communication model, category, and failure models. Alsunaidi [3] also classifies the consensus algorithms into proof-based and vote-based. Another investigation, by Xiao [4], classifies the algorithms into Namakato-based and variations, PoS-based, and other emerging consensus protocols.

Unfortunately, none of the existing studies provides a comprehensive classification that incorporates all classes of consensus mechanisms. This would require identification of prevalent factors so that diversity of the individual approaches can be clearly discerned. The proposed classification not only supports detailed analysis of the building blocks of each algorithm but also provides an extensive guide on the communication model and performance parameters im-



(a) Invevstigated resources on consensus algorithms

(b) Academic articles on consensus algorithms

FIGURE 1: Publication outlets from industrial and academic perspective

portant for algorithm evaluation. To facilitate the consensus selection procedure for future experiments, this study also examines the applicability and reputation of each class of algorithms in different areas of blockchain applications.

The contributions of this survey and the added value it brings in comparison to other reviews in this area can be summarized as follows:

- It provides a comparative review of blockchain as one of the precedent forms of distributed ledger.
- It develops a comprehensive classification of consensus algorithms.
- It reviews an extensive set of 130 consensus algorithms and discerns the classes they are associated with.
- The algorithms in the same class are compared regarding their scalability, finality, adversary tolerance, accessibility, agreement, incentives, centralization, and cost.
- Finally, it analyses the distribution of each class in different blockchain application domains.

To provide a comprehensive analysis of consensus mechanisms, we investigated a wide range of resources from academic journals, industrial websites/blogs, conferences and workshops, to white papers. A total of 185 publications have been selected for consensus analysis. As illustrated in Fig.1, there are two leading publication sources: academic journals (39%) that represent the data gathered from the academic domain, and industrial websites (17%) that denote websites and blogs from the blockchain industry. As further shown in Fig. 3, academic journals are also classified according to their publication date to help identify potential literature gaps.

This article is organized as follows. Section II introduces the distributed ledger technology and its variants. The body of the survey, dedicated to the consensus mechanisms, is contained in Section III and analyzed in Section IV. Final Section V summarizes the main results and conclusions.

II. DISTRIBUTED LEDGERS

The widespread reputation that distributed ledgers have now attained started with the advent of bitcoin cryptocurrency that demonstrated their potency. Their dynamic nature can accelerate transactions and reduce associated expenses by eliminating the requirements for a central authority. DLT is also referred to as a reliant record handling mechanism that tones down cyberattack vulnerabilities through the transparency of transactions. It is a digital database that is constructed, shared, validated, updated, and synchronized by participants to eliminate the single central authority. A distributed ledger is either public (open to all users) or private (shared among particular participants that are more likely to adopt specific protocols during communication). Every participant in a distributed ledger network is required to traverse an agreement phase that differs from one ledger to another and leads to a single decision. A distributed ledger is a secure approach to keep permanent records as it cannot be tampered retroactively. This allows the contributors to communicate through DLT more confidently. The emergence of smart contracts in the context of distributed ledgers as a digital means for verification and execution of contracts has significantly changed the prospect of DLTs. The integration of smart contracts into the distributed ledger has enhanced the reliability, accountability, and transparency of transactional applications. Other than financial transactions, DLTs are applicable to various other use cases.

The Focus Group on Application of Distributed Ledger Technology (FG-DLT) [5], classifies the DLT applications in a horizontal and vertical domain. Vertical domain depicts different sectors of economy and the horizontal domain represents their corresponding use cases [6]. The following subsections provide an overview of the characteristics of the different types of DLT categorized in Fig. 2.

A. BLOCKCHAIN

The notion behind Blockchain as a digital, distributed, and decentralized data structure is the development of transaction blocks that store digital transactions without the need for a central authority. Information concerning new transactions is appended to the chain after it has been encrypted and confirmed by the majority of the participating agents. Each block is then timestamped and cryptographically linked to the former blocks as a demonstration for the sequence of

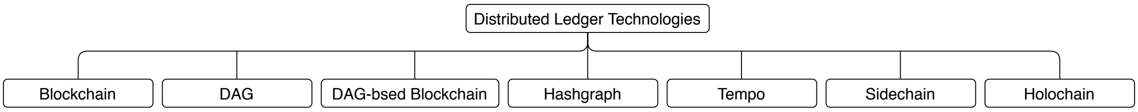


FIGURE 2: Variants of Distributed Ledger Technologies

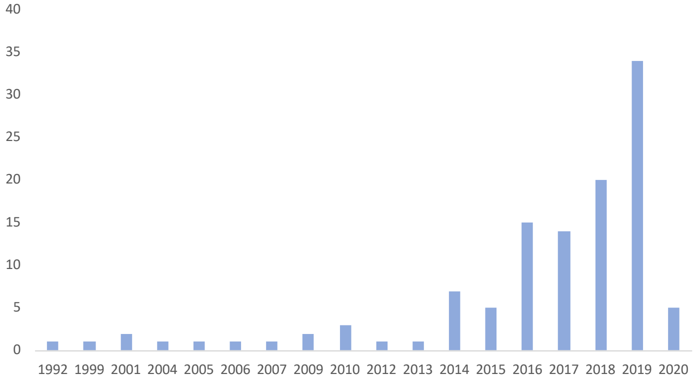


FIGURE 3: Academic journals annual distribution

recorded transactions. As a distributed database, Blockchain comprises an expanding record of transactions accompanied by the chronological order of their occurrence. It keeps the identity of the contributors anonymous by employing digital signatures [7].

1) Immutability

One of the outstanding features of blockchain is immutability, in which no one can modify the distributed ledger by any means. The blockchain remains irreversible since any transaction cannot be altered, deleted, or reversed unless more than 51% of the nodes agree with the modification. This would require the attacker to gain control over more than half of the nodes which is highly improbable. However, although breaching the immutability of the blockchain is considered improbable and complicated, it is possible in practice if a substantial amount of resources is available. The immutability concept is correlated with both data and the code of the distributed ledger. Blockchain considers the immutability of the data records uncontroversial, while data can be tampered and erroneous prior to its appendance to the chain. Although consensus mechanisms are incorporated for verification of the data inputs, they are confined by what participants can faithfully consent to. On the other hand, the immutability of the code is also questioned by pointing out the fact that no code is developed in an impeccable manner integrating all operative requirements. This concern is supported by the fact that, in many cases, the blockchain code has been constantly adapted.

2) Security

Distributed ledgers are recognized for their exceptional security measures. Participating agents utilize cryptography encryption to compose transactions. The public and private keys associated with transacting agents ensure the integrity on top of validation procedures that hinder manipulation. The building blocks of blockchain security are cryptographic hashing functions which generate unique identifiers with regulated length independent of the input. Each hash is associated as an identifier for a block and correlates with the hash value of the former block. The hash function is further utilized in a consensus mechanism for verification of ongoing transactions.

3) Speed

Distributed ledger has overcome the decelerated transactions associated with the classic banking system. Blockchain's transaction speed depends on block size, transaction fee, and network congestion. Blockchain facilitates global transactions by decreasing the block time, which represents the required interval for appendance of a novel block. Moreover, the transmission time decreases with the increase of block size, which improves the transaction speed.

4) Consensus

Consensus mechanisms have been incorporated in blockchains as a fault-tolerant mechanism for transaction verification. The consensus is utilized to preserve agreement among the nodes in the network. When the network expands, the number of nodes increases and it is quite challenging to achieve agreement. Public blockchain requires the participation of users for verification and authentication of the transactions. Since blockchain is a dynamic, self-regulating system, it requires the incorporation of a secure mechanism to ensure the authenticity of the transactions, having participants reach agreement on a consensus. Various types of consensus mechanisms have been proposed that differ in terms of their underlying principles and applications.

B. DAG/TANGLE

Directed Acyclic Graph (DAG) is a variant of DLT that has been proposed as an alternative to blockchain. DAG's co-operating nodes are capable of cross verification due to their arrangement in a directed graph. Implementation of DAG enhances the scalability of the network and reduces the transaction fees as it supports fee-less nano-transactions. DAG significantly improves the transaction validation speed without incentivizing the participants. Since DAG reaches consensus without the implementation of the classic hash-protected PoW (PoW), it requires neither miners nor the underlying energy intensive infrastructure. However, DAG cannot reach a secure decentralized consensus to preserve the security level provided by its counterparts like blockchain. Significant features provided by DAG are listed below:

1) Scalability

DAG is known for its virtually infinite scalability. Unlike other distributed ledgers, DAG enhances the scalability with the expansion of the network. It requires each node to verify at least two former transactions to proceed with the confirmation of their corresponding transactions. Correspondingly, the hashing power required for the validation procedure decreases.

2) Compatibility

Employing microtransactions is avoided in blockchain as it increases the transaction fees. On the contrary, DAG, as a decentralized channel, enables the participants to make instant micro- or even nano-transactions by incorporating transaction fee-free schemes. This feature makes DAG more compatible with microtransactions.

3) Resilience

One of the most significant features of DAG is quantum resistance that makes the underlying distributed ledger less susceptible to quantum computers with higher-level computing properties using Winternitz one-time signature scheme [8].

4) Validation

The quantum resistance of DAG results in masked authenticated messaging and parallelly lined transactions, which is an excellent approach for information transformation through encryption and authentication.

C. DAG-BASED BLOCKCHAIN

Ever since blockchain has been proposed, many investigations have been performed to address the deficiencies of the blockchain, such as its limited throughout. As the next generation of the blockchain, DAG-based blockchain has been proposed. It inherits the significant features of both DAG and blockchain, as discussed earlier. The proposed distributed ledger envelopes transactions in the form of blocks that are structured as DAG. It adopts a verification procedure that requires every novel transaction to be validated by at least two earlier transactions to be appended to the blockchain. This procedure is inherited from DAG as it eliminates the participation of miners for the authentication of transactions, which speeds up the whole process. Moreover, it employs the gossip protocol for communication among nodes, which enhances the linear structure of traditional blockchain [9], [10].

D. HASHGRAPH

Hashgraph emerged as a consensus-oriented scheme using the PoS algorithm. It is capable of storing several transactions in parallel by associating uniform timestamps. Hashgraph incorporates a gossip protocol to transmit transaction information through the network. Each node in the hashgraph selects a random neighbor for information transmission, aggregates all acquired information and passes it to another randomly chosen node. In a short time, all nodes know about the transactions and, using a virtual voting mechanism, each node can validate and append them to the ledger [11].

1) Fairness

Hashgraph allows all contributing nodes to develop signed transactions that are later shared across the network. Since hashgraph employs an enhanced level of fairness, nodes are less likely to be affected by influencers once they agree on a transaction.

2) Data Structure

Hashgraph can be considered a distributed database due to its atomicity, consistency, isolation, and durability properties. This is often referred to as ACID compliance. Once the distributed ledger reaches a consensus on the sequence that each transaction has occurred at, the consensus order will be shared among the nodes' local copy of the database. As the database preserves ACID properties, each node of the ledger as the community can also retain the same features.

3) Information Sharing

As mentioned earlier, hashgraph incorporates a gossip protocol to broadcast transaction information across the network. To ensure all nodes are informed about the changes in the network, the hasgraph ledger keeps a record of each gossip from initiation to termination. This assists hashgraph in inspecting if all nodes have been engaged in the process and are aware of the transactions.

4) Voting

Hashgraph incorporates virtual voting for validation of the transaction by relying on the agreement of at least 2/3 of the network on a particular transaction. It also considers the number of famous witnesses that have contributed to the voting procedure. Famous witness stands for a transaction that has occurred in sequential order and receives the witness's vote. The name comes from the fact that most nodes are aware of its occurrence comparatively faster through the gossip protocol. This accelerates reaching consensus in the same order as the transactions within the network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3065880, IEEE Access

E. TEMPO

Tempo is another variation of distributed ledger that incorporates the same principle of retaining the order of information. In addition, it associates timestamps and subset of the ledger to the users. Once a node begins to verify a transaction, it is not capable of maintaining a consensus using traditional timestamps as the timestamp changes from one sub-ledger to another. Hence, it proceeds by using the logical clock, which compares whether the former transaction matches its recorded sequence.

1) Structure

Tempo is a variation of a distributed ledger that scales horizontally. The data structure and storing mechanism of tempo rely on sub-ledgers perceived as shards. Diverse fragments of shards with distinctive identification keys are the building blocks of a universal distributed ledger. Tempo ensures that all cooperating shards are storing the transaction information in the right order.

2) Communication

As mentioned earlier, tempo is required to ensure that cooperating shards in the distributed ledger are able to store the transaction information in the right sequence. Accordingly, it adopts the hashgraphs random gossip communication protocol that allows all agents to communicate and propagate shared pieces of information. Hence, all shards on the ledger are able to synchronize their information.

3) Verification

One of the most significant features of tempo is associating integer values to agents that increase with the number of unprecedented transactions that the agent observes. This integer number represents the overall number of unique transactions that each agent stores and is referred to as a logical clock. Once the information regarding a transaction is stored by the agent, the subsequent logical clock is stored alongside to facilitate the verification of upcoming transactions.

F. SIDECHAIN

Sidechain is another variation of the disturbed ledger that aims to address the shortcomings of traditional blockchain concerning security, privacy, and performance. The structure of sidechain comprises a combination of two blockchains that control access requests through a central consortium and manage local transactions as a permissioned blockchain. Sidechain fragments the network, which allows each subnet to verify its corresponding transactions and eliminates the scalability challenges caused by acquiring the consensus of all network nodes [12].

1) Privacy and Security

In addition to the significant features of blockchain that have been employed by sidechain, it is capable of preserving the privacy of transactions by keeping specific data confidential from competitor agents in the network. Moreover, it associates each sidechain with a validator node that communicates with the consortium network and verifies local transactions.

2) Performance

The structure of sidechain comprises several linked lists that follow the principle of the blockchain in block generation and reaching consensus. As a private ledger, it has been able to decrease the adversity of PoW and enhance the performance of the network. However, sidechain suffers from overhead generated by the sidechain network and its underlying platform, Monax [13].

G. HOLOCHAIN

Holochain is another variation of distributed ledger, advanced in terms of incorporating an agent-centric structure rather than a data-centric scheme. This type of distributed ledger associates a distinctive forking approach with each agent to eliminate the use of a global consensus mechanism. The motivation behind this approach is to significantly enhance the scalability of the underlying network [11].

1) Validation Structure

One of the significant properties of holochain is that individual modules constitute the distributed ledger, which results in the agent-centric structure. Accordingly, each agent is only in charge of its personal ledger and is not required to execute consensus for validation of each transaction. This not only accelerates the verification procedure but also enhances the energy efficiency in consecutive runs.

2) Explicit Participation

Holochain, as its name suggests, implements a holographic methodology that allows developers to build decentralized applications that inherit distributed ledger features, such as scalability. One of its significant attributes is the explicit participation of the user in the distributed ledger. This endows each user with explicit access to and control over data.

3) Security and Decentralization

Stemming from the properties mentioned above, holochain provides a very high level of security using the DNA scheme [14]. Once a malicious action takes place in terms of broadcasting invalid information, the attacker is required to provide the DNA of the system. Each agent in the distributed ledger is required to validate the DNA of the sender with its own. Holochain is also capable of handling unlimited transactions due to its scalability that results from the agent-centric procedure employed for storing the ledger.

III. CONSENSUS

The notion behind the consensus mechanism is performing frequently secure updates on the distributed ledger. One of the essential techniques is state machine replication that ensures the presence and execution of shared states concerning

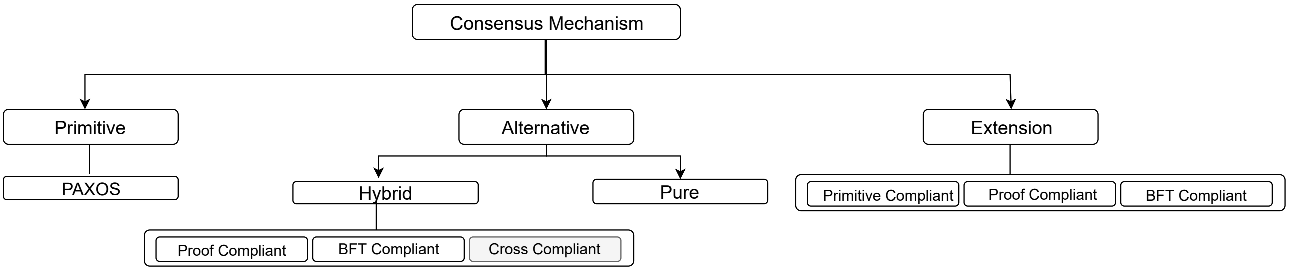


FIGURE 4: Classification of Consensus Mechanisms

the predefined state transition regulations. Since the state is shared among several replicas within the network, the execution of the state will eventually result in identical outputs. Hence, replicas are required to interact and reach agreement on potential modifications of the state using a consensus scheme. Consensus helps replicas to decide on the finality of each state. However, the implementation of consensus in a distributed system is complicated as it requires a consensus mechanism to maintain adversity tolerance, failure resilience, partitioning throughout the network, delay perseverance, and other important properties. Moreover, it involves security measures such as the management of malicious nodes by adopting regulations like synchrony or message broadcast. The significance of consensus in distributed ledgers, including blockchain, is the preservation of three critical properties that ensure the efficiency of the underlying network. Other than maintaining an agreement on a consistent global state for a distributed ledger, consensus ensures the safety, liveness, and fault tolerance of the network. Accordingly, consensus protocols can be evaluated based on these properties. The consensus mechanism can preserve safety if it ensures all nodes will contribute to an identical, consistent, and valid output. Liveness is referred to as the ability of a consensus mechanism to direct the contribution of nonfaulty nodes towards the production of value. The consensus mechanism needs to be capable of recovering from potential failures of contributing nodes to maintain a certain level of fault tolerance [15].

Over the last two decades, there have been introduced various consensus mechanisms. In this contribution, we introduce a novel classification of the most important consensus mechanisms and their variations. The proposed taxonomy is illustrated in Fig. 4 and a more detailed view of the categorization is shown in Fig. 5. We present a comprehensive review of the functionalities, shortcomings, and advantages of the consensus mechanisms guided by the proposed classification and executed using the procedure depicted in Fig. 6.

Accordingly, the first class belongs to the traditional consensus approach paxos that took primary steps towards a fault tolerance mechanism in the presence of unreliable system provisioning. The notion behind this classification can be described as follows. If a consensus mechanism is not a primitive consensus such as paxos, then it is either proof compliant, Byzantine fault tolerance (BFT) compliant, primitive compliant, or cross-complaint. Primitive compliant mechanisms represent all variations of paxos that have been proposed as an extension to this consensus adopting the protocol's main characteristics. Proof protocols and blockchains have been intertwined since the advent of distributed ledgers as they have demonstrated to be effective protocols that improve the audibility and accountability of decentralized networks and preserve their privacy. Moreover, they have been incorporated in the configuration of enhanced proof protocols (proof compliant) or other cryptographic primitives – these are referred to as cross-compliant in the proposed taxonomy. Proof compliances are either extensions to the proof protocols that inherit the main characteristics of the protocol and attempt to enhance the algorithm, or alternatives that have been introduced as novel proof mechanisms. These algorithms address the deficiencies of a particular proof mechanism by incorporating some outstanding features of other proof consensus mechanisms leading to a hybrid proof compliant consensus.

One of the most common faults recognized in distributed ledgers is caused by the erratic behavior of participating nodes, which is known as Byzantine fault. This was first pointed out by Lamport as the Byzantine general problem, which happens as a consequence of a compromised node where a Byzantine node contributes to ambiguous responses or misleads other agents. In response, the BFT algorithm and its variants have been proposed. These are categorized as BFT-compliant algorithms. An algorithm from this category is either proposed as an extension to existing BFT approaches, or as an alternative that integrates the characteristics of BFT protocols and builds upon them to address the corresponding deficiencies. Otherwise, it is considered an alternative that integrates differing BFT protocols and contributes to a hybrid BFT-compliant alternative.

Pure alternatives are associated with protocols that present a novel consensus without preserving the features of previously proposed mechanisms trying to address the existing shortcomings.

A. PRIMITVE CONSENSUS MECHANISM

1) Paxos

As the very first proposed consensus algorithm, paxos facilitates the selection of a single value beneath the crash or faulty circumstances of the network. Paxos classifies the nodes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3065880, IEEE Access

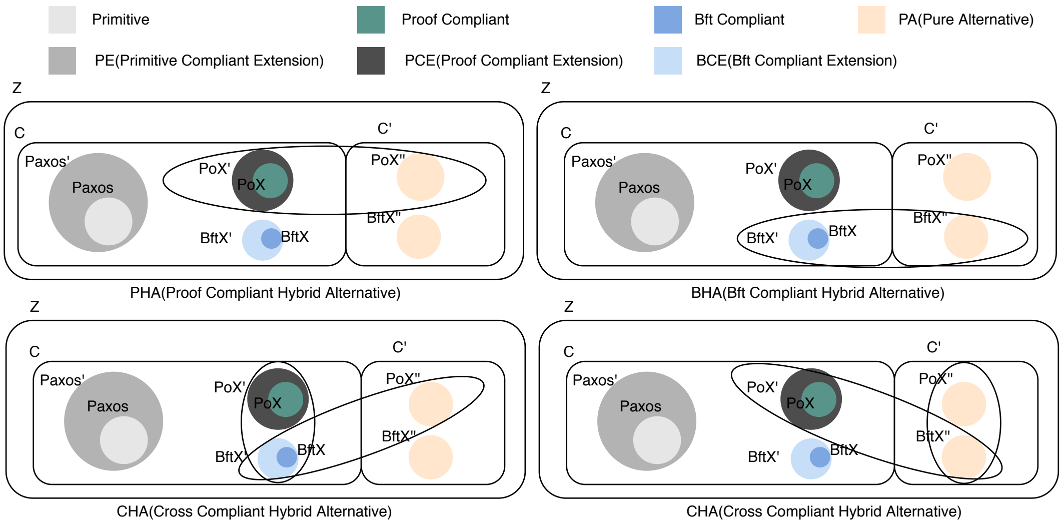

Primitive
Proof Compliant
Bft Compliant
PA(Pure Alternative)
PE(Primitive Compliant Extension)
PCE(Proof Compliant Extension)
BCE(Bft Compliant Extension)
Z
C
C'
Paxos'
Paxos
PoX'
PoX
PoX''
BftX'
BftX
BftX''
PHA(Proof Compliant Hybrid Alternative)
BHA(Bft Compliant Hybrid Alternative)
CHA(Cross Compliant Hybrid Alternative)
CHA(Cross Compliant Hybrid Alternative)

FIGURE 5: Venn diagram for different classes of consensus

TABLE 1: Primitve Consensus Mechanism

Consensus	Scalability	Finality	Adversary Tolerance	Communication Model	Hybrid	Accessibility	Agreement	Incentives	Centralized	Cost
Paxos [16]	Low	Determinstic	N/A	Asynchronous	-	Permissioned	Vote-based	✓	-	↑↑

TABLE 2: Proof-Compliant Hybrid Consensus Mechanism

Consensus	Scalability	Finality	Adversary Tolerance	Communication Model	Accessibility	Agreement	Incentives	Centralized	Cost
PoWS [17]	High	Probabilistic	N/A	Synchronous	Permissionless	Capability-based	✓	✓	↑
PoStakeT [18]	High	Probabilistic	N/A	Synchronous	Permissionless	Capability-based	-	-	Moderate
PoWT [19]	High	Probabilistic	N/A	Synchronous	Permissionless	Vote-based	N/A	N/A	↓
PoSpaceT [20]	High	Probabilistic	N/A	Synchronous	Permissionless	Capability-based	N/A	-	↓
PoD [21]	High	Probabilistic	N/A	Synchronous	N/A	Vote-based	✓	N/A	↓
PoActivity [22]	High	$<$25%	N/A	Synchronous	Permissionless	Vote-based	N/A	N/A	N/A
SnowWhite [23]	N/A	Probabilistic	N/A	Asynchronous	Permissioned	Vote-based	✓	-	N/A
HPoW [24]	Moderate	Probabilistic	N/A	Synchronous	Permissionless	Vote-based	-	-	↑↑
FPoA [25]	N/A	Probabilistic	N/A	Partially Synchronous	Permissionless	Capability-based	✓	-	↓
PoSV [26]	N/A	Probabilistic	N/A	Synchronous	N/A	N/A	✓	-	↑
PoDDoS [27]	N/A	N/A	N/A	N/A	N/A	N/A	✓	-	↑

into proposers, acceptors, and learners. Proposers provide a message indicating a proposal number and forward it to the acceptor. The proposal number is considered as a time line throughout the process, in which the proposal with a higher number is the most recent update. Acceptor compares the acquired proposal number with the current known value, and only accepts the proposal if it is more recent. Afterward, the acceptor forwards a response message indicating whether the proposal has been accepted or rejected, corresponding proposal number, and all accepted value. The proposer is required to investigate whether the majority of acceptors have rejected the proposal or not. In case of rejection, the proposer updates the proposal number with the most recent value. Otherwise, the acceptor broadcasts the accepted value to all learners on the network. To reach consensus in paxos, the proposer should receive at least $N/2-1$ acceptances (N is the number of proposals) from the acceptors [16].

B. PROOF COMPLIANT HYBRID ALTERNATIVE CONSENSUS MECHANISM

1) Proof of Working Stake/Proof of Chain

Proof of chain has been initially employed by CLAIM Coin as an alternative to PoS. PoC is a derivative of PoS that enhances network security with distribution and transparency. This approach incentivizes staking users by choosing an active client within 1-minute intervals. The client verifies all pending transactions associated with the CLAMS network. This process eliminates the proportional incentive mechanism of the former PoS. PoC encourages the participants to actively execute their CLAM, which results in improvement of network security. The notion behind the distribution of CLAMS is to engage as many contributors as possible to not only facilitate the authenticity verification process but also to make the network widely spread and difficult to track [17].

2) Proof of Stake Time

Proof of stake time (PoST) is a time-accepted nonlinear consensus mechanism that has been proposed as an alternative to address the deficiencies of PoS. PoST incorporates a periodic time acceptance function that correlates with retained coins and strives to enhance the security and distribution of the network. The contribution of volunteers is determined using

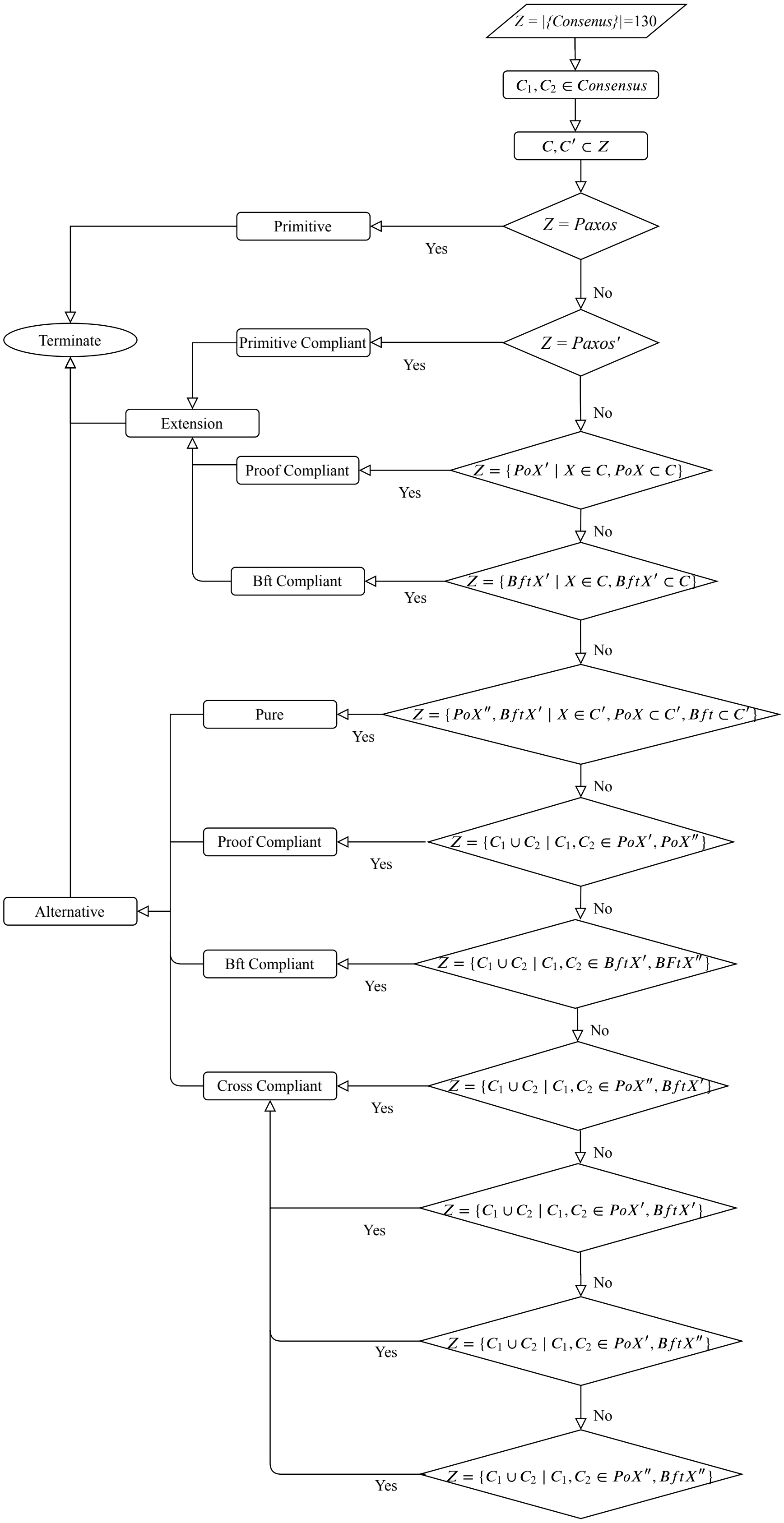

$Z = |\{Consenus\}| = 130$
$C_1, C_2 \in Consensus$
$C, C' \subset Z$
$Z = Paxos$
Yes
Primitive
No
Terminate
$Z = Paxos'$
Yes
Primitive Compliant
No
Extension
$Z = \{PoX' \mid X \in C, PoX \subset C\}$
Yes
Proof Compliant
No
$Z = \{BftX' \mid X \in C, BftX' \subset C\}$
Yes
Bft Compliant
No
$Z = \{PoX'', BftX' \mid X \in C', PoX \subset C', Bft \subset C'\}$
Yes
Pure
No
$Z = \{C_1 \cup C_2 \mid C_1, C_2 \in PoX', PoX''\}$
Yes
Proof Compliant
Alternative
No
$Z = \{C_1 \cup C_2 \mid C_1, C_2 \in BftX', BFtX''\}$
Yes
Bft Compliant
No
$Z = \{C_1 \cup C_2 \mid C_1, C_2 \in PoX'', BftX'\}$
Yes
Cross Compliant
No
$Z = \{C_1 \cup C_2 \mid C_1, C_2 \in PoX', BftX'\}$
Yes
No
$Z = \{C_1 \cup C_2 \mid C_1, C_2 \in PoX', BftX''\}$
Yes
No
$Z = \{C_1 \cup C_2 \mid C_1, C_2 \in PoX'', BftX''\}$
Yes

FIGURE 6: Flow chart of Consensus classification procedure

an interest rate that maintains inverse proportion to network strength. PoST defines a quantified idle-time attribute to represent a fraction of age that does not reinforce the distribution of the consensus anymore. This parameter affects the fraction of acquired interest and eliminates the probability of meeting the proof. Hence, to enhance the interest rate, the participating node is required to stake constantly to pass all corresponding nodes through stake-time window [18], [28].

3) Proof of Work Time

Discovering the target nonce in PoW, a waste of computational power seems inevitable as a consequence of regulated intervals for block creation. As a PoW alternative, PoWT consensus incorporates a block time attribute to enhance not only the mining power but also the transactions of the blockchain. It conforms the speed of the transactions to mining power and facilitates auto-adjusting towards profitable mining. This scheme proposes a variable block creation rate that correlates with increments of mining power that simply eliminates the waste of required computation power [29].

4) Proof of Space Time

Proof of space-time (PoST) is another implementation of the proof of storage in which the server can publish a proof of storage where verifiers can refer to and investigate whether the transmitted data was being stored during a particular period of time. Proof of space time eliminates the submission of proofs to the blockchain that correspondingly prevents frequent interactions of the verifier and prover. Hence, miners are entrusted to store the user's data in exchange for a collateral deposit. Afterward, the miner will store data within the duration indicated in the settlement and submits the corresponding PoST to the network as an evidence [20], [30], [31].

5) Proof of Devotion

Proof of devotion (PoD) is an integration of PoS and PoI in which nodes with the greatest impact on the network's ecology are given permission for block generation. Block proposer will be chosen using participants from a division of validator sets, which are referred to as dynasties in a round of BFT voting. This process facilitates determining the legitimacy of the proposed block. Moreover, in order to eliminate the titled probability that may cause a monopoly, PoD grants bookkeeping titles to the designated nodes [24], [21], [32].

6) Proof of Activity

Proof of activity has been proposed as an alternative for PoS that integrates both PoS and PoW to incentivize the participating agents instead of penalizing the passive agents. Proof of activity exploits the hash of the most recent block to select pseudorandom stakeholders for validation of a recently mined block template. Each block will be appended to the blockchain after being validated and having its hash signed by the stakeholders [22], [33].

7) Snow White

Snow white proposes a consensus mechanism known as sleepy. It incorporates the same procedure as the proof of activity for the election of the committee nodes that are in charge of voting on the block generation leader. Snow White ensures to select a qualified leader concerning the hash function. Snow White differs from PoW in terms of feeding the hash function with timestamps, which is an alternative for the arbitrary nonce. Finally, Snow white can bear successive committee reconfigurations and fraud nodes [23], [34].

8) Hybrid Proof of Work

Hybrid proof of work, which was initially introduced by Lynx, is a PoW alternative that maximizes the contribution of miners with limited computational resources by eliminating profit incentives. HPoW does not necessarily grant the reward to the fastest node since it randomly selects the candidate blocks without imposing any requirements on the hashing power or speed. Moreover, it does not allow a single miner to successively win a block in 30 minute time intervals [35], [36] [24].

9) Flexible Proof of Activity

Flexible PoA is a hybrid consensus mechanism that employs PBFT to address forking and provide integrity. Flexible PoA employs a rotating committee that selects miners with respect to their PoW and PoS power and capabilities. The proposed fork-free mechanism is a generalized PoW variant that enhances the evaluation procedure of the hash function and contributes to the exclusion of selfish mining [25].

10) Proof of Stake Velocity

Considering the major drawbacks of PoW and PoS, proof of Stake Velocity (PoSV) has been proposed as an underlying consensus algorithm for Reddcoin to facilitate secure peer-to-peer (P2P) transactions. PoSV integrates both stake and velocity to decrease the mining waste and prevent multipool threats by incorporating novel coinage functions. In comparison with PoS protocols that consider coin age linear, PoSV exploits a monotonic decay function for coinage estimation that significantly changes the incentives and results in the exceeding rate of coinage accumulation for the most recent coins. The exponential decay function imposes an asymptotic restriction on coinage for security means that makes the execution of 51% attacks extremely difficult [26].

11) Proof of DDoS

Proof of DDos is the underlying consensus mechanism for DDosCoin and operates by incentivizing the participants that attack victimized servers by transmitting a large amount of network traffic. This approach is a PoW alternative, also known as malicious PoW. It requires numerous TLS connections between the miners and the target server. The target server is fairly selected running a PoS consensus that ensures the participants hold a substantial stake from mining activities. Proof of DDos can also be incorporated to measure

TABLE 3: BFT-Compliant Hybrid Consensus Mechanism

Consensus	Scalability	Finality	Adversary Tolerance	Communication Model	Accessibility	Agreement	Incentives	Centralized	Cost
LFT [38]	High	Deterministic	N/A	Asynchronous	Permissioneless	Vote-based	N/A	N/A	N/A
PBFT [39]	Low	Deterministic	$< 33.3\%$	Synchronous	Permissioned	Vote-based	N/A	-	N/A
hBFT [40]	High	N/A	N/A	Partially Synchronous	N/A	Vote-based	N/A	N/A	↓

the bandwidth utilization or resource consumption of the target [27], [37].

C. BFT COMPLIANT

1) Loop fault Tolerant

ICON is an interchain proposing an underlying ecosystem to connect a wide range of blockchains preserving their consensus algorithms. This platform enables data sharing mechanisms among different associations such as universities, banks, or any other private blockchains without imposing any requirements for intermediaries. ICON is a variation of tendermint [41] that reaches consensus using an enhanced BFT known as loop fault tolerance. LFT speeds up the consensus procedure and eliminates forking using a group of trusted nodes that are allowed to regulate the number of required votes. LFT incorporated spinning to facilitate the complications of selecting the primary node. Moreover, it reaches consensus based on message relay and allows a limited number of nodes to generate a block while the remaining nodes participate in voting. This way, the communication overhead is eliminated due to the integration of messages from the network. LFT has successfully decreased the BFT's traditional 3 stage execution procedure to 2.5 stages in which a certain number of nodes are associated with a block generator broadcast at the same time that the rest of the network contributes to the voting procedure [38], [42].

2) Practical Byzantine Fault Tolerance

PBFT has been proposed as a promising solution for Byzantine faults. Several approaches have accordingly adopted PBFT as their underlying consensus mechanism. PBFT proceeds by determining a novel block during each round to classify transactions concerning their sequence. Each node will be passed through 3 different phases known as pre-prepared, prepare and commit, if it has been verified by at least 2/3 of the co-operating nodes in the blockchain network. Other than blockchain, hyperledger is one of the DLT solutions that incorporates PBFT since it can address over 1/3 malicious replicas. As a permissioned and network-intensive consensus mechanism, PBFT ensures the security of the ongoing transactions among acknowledged participants, but it is not able to scale to large networks. This makes it a perfect fit for private blockchains [39], [22].

3) hBFT

hBFT is a hybrid, leaderless BFT variation that incorporates PBFT's checkpoint mechanism which enables the detection of potential inconsistencies in replicas during the message exchange phase. hBFT narrows down the required number of cryptographic procedures and implements speculation to increase performance and resilience while reducing the operation costs. Although hBFT is capable of tolerating any number of faulty clients, it cannot refrain from its significant effect on performance [40].

D. CROSS COMPLIANT HYBRID ALTERNATIVE CONSENSUS MECHANISM

1) Combined Delegated Proof of Stake and Byzantine Fault Tolerance

The integration of DPoS and BFT, which was initially introduced by Credits [49], incorporates both algorithms for different stages of the consensus mechanism. To decide on the contributing agents, DPoS is applied to the network. Selected nodes contribute to the transaction verification procedure and append a block that comprises a sequence of authentic operations into the blockchain. BFT is in charge of updating the ledger and confronting potential security threats. The participation of agents in each round requires transmission of the hash of the most recent block to its corresponding generator within a particular interval. Failing to do so would eliminate the node from participation in the corresponding round. Moreover, nodes that provide the correct hash value are stored as authenticate nodes within a chronologically ordered list [43], [50].

2) Casper

Casper is an alternative to PoS consensus algorithm that integrates the BFT mechanism. This approach incorporates dynamic validator sets along with a correct-by-construction forking scheme in which validators are required to vote and broadcast their signed votes throughout the network. Casper typically adds PoS on top of PoW as a supplementary layer to ensure finality. This also enhances the network's modular overlay. Casper aims to improve the security issues associated with PoS like long-range revision attacks. However, the evaluation results show that Casper is not able to tackle 51% of attacks [44].

3) BFT-Raft

BFT-Raft has been proposed as an alternative to the classical raft. It integrates the features of both BFT and raft, including security and fault tolerance characteristics. To ensure the authenticity of the messages, BFT-raft exploits digital signatures to preserve the integrity of messages and eliminate forging. Messages that convey invalid signatures can be easily acknowledged and outcasted since they are signed by both nodes and users. BFT-raft elects leader nodes through the voting procedure and is capable of sustaining networks functionality along f Byzantine failures in the presence of at least $3f + 1$ nodes in the network [45], [51].

TABLE 4: Cross-Compliant Hybrid Consensus Mechanism

Consensus	Scalability	Finality	Adversary Tolerance	Communication Model	Accessibility	Agreement	Incentives	Centralized	Cost
CDPoS-BFT [43]	Very High	Probabilistic	N/A	N/A	Permissionless	Vote-based	✓	-	↓
Casper [44]	High	N/A	$<51\%$	Synchronous	Permissionless	Vote-based	✓	-	↑
BFT-Raft [45]	N/A	Probabilistic	N/A	Partially Synchronous	Permissioned	Vote-based	N/A	N/A	↑
Peercensus [46]	Low	Deterministic	N/A	Synchronous	N/A	Vote-based	✓	-	N/A
VBFT [47]	High	N/A	Immediate	Synchronous	Permissionless	Vote-based	N/A	N/A	N/A
DDPoS [48]	High	N/A	N/A	Asynchronous	Permissionless	Vote-based	✓	-	↓

4) PeerCensus

PeerCensus is a hybrid consensus mechanism that aims to dissociate the block creation procedure from transaction verification. It inherits the functionalities of both PoW and BFT to grant voting and block generation rights. PoW mechanism is employed for the election of the block leader while block validation proceeds through BFT's pre-prepare, prepare, and commit stages. Miners committee is arranged concerning the miners whose preceding mined blocks reach a certain level of depth in the chain. Hence, the transaction that has been approved through BFT does not require any further mining process [46], [34].

5) Verifiable Byzentine Fault Tolerance

VBFT is an alternative consensus mechanism for BFT that has been initially adopted by the Ontology Consensus Engine (OCG). It integrates PoS, BFT, and the Verifiable Random Function(VRF). The OCG's consensus network comprises consensus nodes in charge, preserving the blockchains balance. Incorporation of VRF facilitates the consensus population generation, as it provides randomness in the selection of nodes, whether they are proposers or verifiers. This not only enhances the resilience but also accelerates the finality of the consensus mechanism [47], [52], [53].

6) Delegated proof of stake with downgrades

Delegated proof of stake with downgrades (DDPoS) is a consensus mechanism inspired by PoW and DPoS. DDPoS minimizes the implications of stakes and computing resources on generating blocks. Using the downgrading mechanism, DDPoS is able to replace malicious nodes and enhance the security of the system. The performance evaluation results of DDPoS indicate that it outperforms PoW and PoS in terms of efficiency. However, its performance is still slightly lower than DPoS [48].

E. PURE ALTERNATIVE CONSENSUS MECHANISM

1) Zab

Zab has been proposed as an atomic broadcast protocol that incorporates primary ordering, which is significant in the context of primary back-up systems. Primary ordering is responsible to ensure the validation of state alteration orders over time and course of transactions. The validation procedure confirms the incremental order of each state change with respect to the preceding one. This approach maintains an implicit interdependence on the sequence of state alterations. For the identification of misplaced or missing alterations, zab exploits a transaction identification method on state alterations to conduct an initial determination on which order of transactions can be employed to retrieve the application state . Zab was originally implemented on the ZooKeeper [95] and has proven to be adequate for web-scale applications. It outperforms paxos, performing thousands of broadcasts per second [54].

2) Proof of Work

The proof-of-work mechanism commences with the calculation of the hash value of the block header. The block header comprises a nonce that is frequently modified by miners to obtain different hash values. Hence, the consensus necessitates the earned value to remain within a particular bound. To maintain an agreement across the network concerning the propagation of new blocks, PoW imposes a complicated puzzle that needs to be resolved by cooperating nodes. Miners that overcome the puzzle will be granted access to append a new block. The puzzle preserves an adjusted difficulty and is confronted by estimating the nonce's value. This value is incorporated with the block's header information to feed the SHA-256 hash function. The hash function will then convert all inputs to generate the hash value. If the output of the hash function holds a value beneath an appointed threshold, the estimated nonce will be accepted, and the miner is allowed to append a block into the blockchain. Hence, whenever the objective value is acquired by a miner, it will accordingly broadcast the block throughout the network, and every single node within the entire network will be asked to confirm the authenticity of the hash value and append the corresponding block to their blockchain [55].

3) Proof of Luck

Proof of luck has been originally built upon TEE [96] and XGS [97]. With the proof of luck consensus algorithm, the procedure of appending a new block to the miner's chain requires allocating a luck assess to each block. The luck assess is a random number that ranges from 0 to 1 within a uniform distribution, and the chain that retains the overall largest assess is consented to be considered as the main chain. The chain that maintains the highest luck is preferred by the blockchain. These characteristics make the proof of luck resistant to the double-spending attack. However, the proof of luck suffers from power deficiency since it decides on the luck access after several examinations. Moreover, unsynchronized timepiece between the node and the network may eliminate the chances of the node for being lucky, which marks the significance of executing the proof of luck after synchronization of miners [56].

TABLE 5: Pure Alternative Consensus Mechanism

Consensus	Scalability	Finality	Adversary Tolerance	Communication Model	Accessibility	Agreement	Incentives	Centralized	Cost
Zab [54]	High	Determinstic	N/A	Asynchronous	Permissioned	Vote-based	✓	-	↑↑
PoW [55]	High	Probabilistic	<=25%	N/A	Permissionless	N/A	N/A	-	↑↑
PoL [56]	High	Determinstic	<50%	Asynchronous	Permissioned	Capability-based	✓	-	↑
PoE [57]	Moderate	Probabilistic	N/A	N/A	N/A	Capability-based	✓	-	↑
PoB [58]	High	Deterministic	<51%	N/A	Permissionless	Vote-based	-	✓	↓
PoT [59]	High	Probabilistic	<25%	N/A	Permissionless	N/A	N/A	-	↑
PoSpace [60]	High	Probabilistic	<25%	N/A	Permissionless	Vote-based	N/A	-	↓
PoExistance [61]	N/A	Deterministic	N/A	Synchronous	Permissinless	N/A	✓	-	N/A
PoM [62]	High	N/A	N/A	N/A	Permissionless	Vote-based	✓	-	N/A
PoAu [63]	N/A	Deterministic	N/A	Synchronous	Permissioned	Vote-based	N/A	-	N/A
PoAp [64]	N/A	Probabilistic	N/A	Partially Synchronous	Permissionless	Vote-based	✓	-	↓
PoKH [65]	N/A	N/A	N/A	N/A	Permissioned	Vote-based	N/A	-	N/A
PoI [66]	N/A	N/A	N/A	Synchronous	Permissionless	Capability-based	✓	✓	N/A
PoCredit [67]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PoPlay [68]	Very High	Deterministic	51%	Synchronous	Permisisonless	Vote-based	✓	-	↓
PoF [69]	N/A	N/A	N/A	N/A	N/A	Vote-based	N/A	-	↓
Flash Consensus [70]	N/A	Deterministic	N/A	Synchronous	N/A	Vote-based	-	-	N/A
PoCooperation [71]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Obelisk [72]	High	Deterministic	50%	Synchroous	Permissionless	Capability-based	✓	-	N/A
PoValue [73]	High	N/A	N/A	N/A	Permissionless	Vote-based	✓	-	N/A
PoDisintegration [74]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PoLearning [75]	N/A	Probabilistic	51%	N/A	Permisisonless	Vote-based	✓	-	↓
PoEligibility [76]	N/A	N/A	<30%	Asynchronous	N/A	Vote-based	N/A	✓	N/A
PoRep [77]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PoVote [78]	N/A	Deterministic	<50%	Synchronous	Permissioned	Vote-based	✓	-	↓
PoPF [79]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PoIndividuality [80]	N/A	Probabilistic	N/A	N/A	Permissionless	N/A	Vote-based	N/A	↑↑
PoPersonhood [81]	High	Deterministic	N/A	N/A	Permissionless	Vote-based	✓	-	↓
SIEVE [82]	N/A	Probabilistic	N/A	Partially Synchronous	Permissioned	Capability-based	N/A	-	↓↓
PoStake [83]	High	Probabilistic	<51%	Synchronous	Permissioned/less	Vote-based	✓	-	↑
PoET [84]	High	Probabilistic	Unknown	N/A	Both	Vote-based	N/A	-	↑
Ripple [85]	Low	Deterministic	<20%	Synchronous	Permissionless	Vote-based	-	-	↑
PoL [86]	High	N/A	N/A	Synchronous	Permissionless	Capability-based	N/A	-	N/A
PoCredibility [87]	N/A	Probabilistic	N/A	N/A	Permissionless	Vote-based	N/A	-	N/A
PoHistory [88]	High	Deterministic	N/A	Synchronous	N/A	Capability-based	✓	✓	N/A
PoEnergy [89]	High	Probabilistic	N/A	Asynchronous	Permissioned	N/A	✓	-	↓
PoWP [90]	N/A	Probabilistic	N/A	Synchronous	Permissioneless	Vote-based	✓	-	↑
PoO [56]	High	Deterministic	N/A	Synchronous	N/A	Capability-based	✓	-	↓
Bitcoin-NG [91]	High	Deterministic	50%	N/A	Permissioned	N/A	✓	✓	↓
PoAsset [92]	High	Deterministic	N/A	synchronous	N/A	N/A	✓	-	N/A
PoBelievability [93]	N/A	Probabilistic	N/A	N/A	Permisisoned	Capability-based	N/A	-	N/A
PoT [94]	High	Deterministic	N/A	Asynchronous	Both	Vote-based	N/A	-	↓

4) Proof of Burn

Proof of burn has been proposed as an energy-efficient, sustainable alternative to PoW where miners use an irretrievable address to convey coins and burn them. The irretrievable address is referred to as eater address and incorporates a public key that is not correlated with any private key to prevent coin retrieval. Once a coin is sent to the eater address, it is permanently eliminated from the network. In PoB, the miners do not invest in physical currency as the cryptocurrencies are burned intentionally to denote the investment in blockchain. Burning cryptocurrency generates virtual mining power; hence, the more coins a user burns in favor of the system, the more mining power it acquires. In addition, the miner, as mentioned above, is more likely to be appointed as the validator of the block. The notion behind PoB is similar to PoW as getting the title of mining a block in PoB bears a resemblance to purchasing computing resources in PoW. All transactions that indicate transferring coins to eater addresses are recorded, and SHA-256 is employed for the calculation of the burn hash concerning each transaction in the network. Eventually, the miner who holds the least assess of the burn hash attains the mining right [58], [98].

5) Proof of Excellence

Proof of excellence is an unexplored conceptual solution to the distribution problem that has been originally introduced in the PoS white paper [57]. This approach incorporates intermittent tournaments and the performance of the contributors in the tournament, which imitates the real-life tournament rewards. A game is employed to select the node that keeps the consensus of the blockchain and establishes an inequitable pattern that allows competent players to write blocks frequently. As a consequence, the blockchain will turn into a partially centralized platform controlled by good players [57], [68].

6) Proof of Time

Considering the computational deficiency of PoW, this approach cannot be adopted in various range of applications, including electric vehicles that maintain more confined processing capability. Hence, the proof of time is an alternative that prevents spanning and Sybil attacks by exploiting the time difference between the two transactions. Moreover, clients are obliged to collect random tokens which make the potential attacks costly as the intruder needs to outpace the throughput of each transaction, along with their associated timestamps [59], [56], [99].

7) Proof of Space / Proof of Capacity / Proof of Storage

Proof of space is a consensus algorithm that is cheaper than PoW in terms of required computing infrastructure, as it requires the use of hard disks or cloud storage systems. It operates based on large stored data sets, known as parcels. Their multiplicity enhances the probability of mining a new block for the corresponding node. The proof of space is executed in two stages. The first stage is plotting, where the hard disk capacity that miner has devoted is evaluated by incorporating Shabal [100] hash function and plotting the hard disk. The hash function is then seeded using miner's ID and nonce [60]. Mining is performed during the second stage. It refers to the most recent block on the chain to calculate the generation hash. The total number of scoops is then calculated by incorporating the hash module to generate the target value that also uses the outputs of the plotting stage. Afterward, the network re-calculates the scoop for each hash to validate the deadlines for each miner. The miner that correlates the shortest published deadline generates the next block and receives a reward for the transaction. The advantage of PoSpace is its energy efficiency, as it does not impose high requirements on hardware. However, it is known to be susceptible to malware attacks, since its hashes are stored on a hard disk. This enhances the vulnerability of the data and risks of being tampered with. Spacecoin [101], Chia [102], and Burstcoin [103] are a few of the many approaches that have incorporated PoSpace [104], [43]. Other names used for this consensus mechanism are Proof of Capacity and Proof of Storage.

8) Proof of Existence

Proof of existence as an online service exploits a decentralized certification SHA256. PoE permanently validates the existence of data by storing its cryptographic digest and the corresponding submission date using blockchain. This service can publicly prove the ownership of data without revealing the data itself. It also eliminates the requirements for trusting any central authority. This approach provides anonymity, privacy, and decentralized proof that does not rely on a single centralized entity. The application of PoE ranges from ensuring the integrity of documents, document time stamping, and denoting the ownership of data without disclosing the content. PoE, as a blockchain notary service, provides instant and secure validation of the existence of any document, agreement, or contract. Moreover, it implies rules to allow updates in the documents and keeping track of the updates [61], [105]. Factom [106] is an example of approach that incorporates proof of existence.

9) Proof of Movement

Proof of movement has been proposed as an innovative consensus that incentivizes road miners to run Lazooz Dapp [107] on their smart devices. Lazooz encourages the miners to participate in sharing their transportation data and to assist Lazooz to eliminate meandering by weaving the social transportation web. Participants receive incentives in the form of tokens that are acknowledged as "zooz" and can be exploited for transportation and ride-sharing services. The number of collected tokens correlates with the traveled distance. Moreover, Lazooz integrates several algorithms to facilitate decision-making procedures in the absence of user intervention, monitoring the utilization of particular districts to enhance services in accordance with the region's active contributors. Lazooz is a decentralized scheme that allocates weight to miners conforming to frequent crowdsourcing since conventional decisions are made by participants. Proof of movement has been introduced as an alternative to commonly used consensus algorithms. This algorithm incentivizes users to share their transportation information on Lazooz Dapp for the social transportation web. It is a decentralized autonomous platform that makes official decisions using the collective mechanism. As such, it can be used for decision-making procedures that require the elimination of human interventions [62], [108].

10) Proof of Authority

Proof of Authority has been proposed as an underlying consensus algorithm for permissioned blockchains. This algorithm substitutes a lighter message transmission scheme in comparison with BFT algorithm, which has led to the superiority of this approach concerning its performance. There are two implementations of the PoA algorithm known as Aura and Clique [109], which was primitively implemented on ethereum for private networks. Both Aura and Clique exploit a similar block proposal scheme in which the trusted authority such as the current mining leader proposes a new block. Afterward, Aura performs the block acceptance procedure, which is not required in Clique implementation. PoA is executed in several time divisions and during each interval; the authorities alternate using round-robin to propose blocks. Each proposed block is accepted once its signed off by the majority of authorized entities. Moreover, the procedure of discerning authorities results in the centralized configuration of PoA, which makes this approach appropriate for private consortiums [110], [63].

11) Proof of Approval

Proof of approval is acknowledged as a permissionless consensus that intermittently publishes blocks within predefined intervals. Each node can propose a new block; however, nodes that do not indicate valid transactions are eliminated, and stakeholders with a minimum stake are authorized to compete in the block creation procedure. Once the block generator has been selected, it broadcasts its corresponding approval block containing the acquired confirmations and is rewarded with the transaction fees of the proposed block and coins [64].

12) Proof of Know How

Since the adaption of blockchain in the context of standardization, communities can significantly enhance the development procedure of standards. Proof of know how (PoKH) is an underlying consensus algorithm for blockchain-oriented standard drafting that incorporates KH now as tokens. This consensus will ensure the each proposed piece of guidance will be implemented at least once prior to being appended to the standard. Each participant will be incentivized to use PoKH concerning the quality and quantity of proved tests. Moreover, PoKH will make sure to close each block after appending the implementations that exemplify guidance. This approach will affect market intake and standard adoption through practical exemplifications [65].

13) Proof of Intelligence

Proof of intelligence has been introduced as an underlying consensus mechanism for high-level blockchain-oriented smart networks beyond cryptocurrencies and ordinary transactions. It can be incorporated as an economic assess to prevent denial of service attacks. Moreover, it can be considered as a reputation qualifier in the context of machine learning or neural network training [66].

14) Proof of Credit

NULS project has proposed an opensource blockchain infrastructure that incorporates cross-chain technologies to facilitate the development of applications. NULS provides the required building blocks and allows developers to employ the consensus of their choice. However, the main chain executes the Proof of Credit consensus, which has been proposed as an alternative to PoS as it requires to lock a certain number of tokens prior to the execution of any node on the network. Moreover, the proof of credit incorporates the master node approach in which the agent node that is considered the owner is more likely to be incentivized from transaction fees. It ensures the stability and integrity of cooperating nodes [67], [111], [112].

15) Proof of Play

In the context of blockchain P2P games, coping with transaction costs and latency is a significant challenge. Previously proposed consensus algorithms such as PoW have become the bottleneck when applied to P2P blockchain games. Hence, Proof of Play has been proposed as a decentralized approach to address the compatibility issues using a seamlessly integrated blockchain in P2P games. This consensus overcomes the data storage issues by simply creating a consensus using the blockchain itself, which prevents modification of the game for the use of blockchain. The notion behind PoP is to incorporate the user's cognitive cooperation in the mining procedure as both miner and user. This incentivizes the procedure by changing the users' perception from running the blockchain to playing the game [68].

16) Proof of Foundation

Nexty [113] is an implemention of a blockchain platform that performs immediate transfers at zero cost. Nexty exploits a Dual Cryptocurrency Confirmation System (DCCS) along with Proof of Foundation (PoF) consensus to eliminate transaction delays and spamming. Proof of foundation is a consensus algorithm that is incorporated within a transaction confirmation protocol and generates the mining reward concerning the computing power that has been utilized during transaction confirmation. Proof of foundation outperforms the commonly used consensus algorithms in terms of associated transaction fees and throughput [69].

17) Flash Consensus

Flash consensus (FC) exploits representatives to maintain consensus. This allows participants to use FLASH coins to vote for representatives concerning the transaction fees. Instead of incorporating a common PoW or PoS, FC allocates a time slot to each miner that allows them to perform the mining procedure. This will facilitate ensuring the sequence of block generation rights. In order to meet the consensus and authorize a block, FC defines a consensus height parameter that is reached when more than 50% of the miners have appended a block to any given block on the chain [114], [70].

18) Proof of Cooperation

Proof of cooperation has been implemented by Faircoin [115] in which every single node follows the same regulations to preserve the integrity and security of the network. In order to maintain the integrity of the cooperatively validated nodes (CVNs), each node is authorized using P2P proof of cooperation. It facilitates the collection and transmission of assets. Moreover, PoC validation mechanism has proven to be efficient since no matter how many CVN's have authenticated the proof, only one mutable signature is required for the appendance of the block into the blockchain [116].

19) Obelisk

Obelisk consensus mechanism has been initially employed in SkyCoin [72] that proposes a fully scalable protocol for decentralized services. To prevent decentralization and its corresponding issues, SkyCoin leverages nodes that are subscribed to a list of trusted nodes. Nodes that maintain greater subscriptions are given more credit in terms of influencing the network. Since obelisk is not dependent on the mining incentives, it does not suffer from the vulnerabilities of the commonly employed consensus mechanisms [117] .

20) Proof of Value

Proof of value has been introduced as an alternative to proof of replication. It regulates the discerned values of different participants exploiting a P2P evaluation mechanism. To maintain consensus, it assigns influence weights concerning the participating value and the acquired alignments with the overall perception of the value. PoV concentrates on the human contribution and incentivizes the active participation with respect to community values. This approach alters perspectives from algorithms to human intervention [73].

21) Proof of Disintegration

B3Coin [118] implements proof of disintegration as an underlying consensus mechanism to address the drawbacks of Proof of burn. Using PoB, coins are burned once sent to an irretrievable address. To eliminate the circulation caused by the proof of burn, PoD destroys the corresponding coins to make them irrecoverable and eliminates the additional circulation and supply. Proof of disintegration is applied on the nodes known as fundamental nodes that are more likely to be incentivized for staking [74], [24].

22) Proof of Learning

Proof of learning is introduced as an underlying consensus mechanism for WekaCoin. To decrease the computational loss affiliated with hashing puzzles, proof of learning incorporates a distribution of classified machine learning schemes as a verifiable database. Hence it implements machine learning contest for verification of the transactions within the blockchain. The term "machine learning competition" refers to the crowdsourcing methods where participators are incentivized for performing published tasks. Proof of learning was initially inspired from reCAPTCHA [119]. It proceeds to line up transaction validation with classification and storage of machine learning approaches towards the development of a public distributed database [75].

23) Proof of Eligibility

BFCV (Byzantine fault tolerance connected vehicles) incorporates an underlying consensus mechanism based on proof of eligibility. This approach reaches Byzantine agreement exploiting the unique features of each node (vehicle) and eliminates the attempts of irrelevant nodes for cooperation in a consensus mechanism. Eligibility of cooperating nodes is evaluated based on qualities such as the presence of a node (vehicle) within the vicinity of the information source. Proof of eligibility accelerates the consensus procedure in a distributed manner. The performance evaluation results indicate the superiority of this algorithm among information dissemination schemes [76].

24) Proof of Reputation

Proof of reputation is the underlying consensus mechanism for the blockchain network that generates a reputation for each node regarding its assets, transactions, and contributions in the consensus procedure. Proof of reputation consists of 3 main stages, including leader selection and block generation, reputation-oriented consensus, and finally updating the reputation values. Once a leader node proposes a new block, it is evaluated through reputation-oriented voting. The node retaining the highest reputation value is validated, and its proposed block is verified accordingly. Nodes that maintain higher reputation values are involved in the voting process, and each node is incentivized regarding its preserved reputation value. The competency of the proposed scheme is highly dependent on the leader selection since the voting consensus of a highly reputed node enhances the security of the protocol [120], [77].

25) Proof of Vote

Proof of vote has been proposed as a PoW alternative for consortium blockchain. The consensus is reached through a decentralized voting arbitration among consortium participants. Four security identities are designated for participants to pursue the voting mechanism. Using the proof of vote, submission or validation of the generated blocks does not require the intermediary of third parties. In comparison with PoW as a fully decentralized approach, this scheme follows a discrete voting and executive principle to decrease the transaction verification time, enhance the convergence, reliability and security [78].

26) Proof of Participation and fees

Proof of participation and fees consensus algorithm has been proposed as an alternative to PoW, initially employed on JCLedger. PoPf performs mining procedure exploiting the contribution of candidates that are selected in accordance with two factors: the fees that the participant has paid and the participation intervals. PoPf ensures that users with a constant contribution to JCLedger are given the chance of being an accountant. Performance evaluation of PoPf indicates the efficiency of this approach when compared with PoE in terms of ensuring the computing power efficiency without imposing security threats [79].

27) Proof of Location

Proof of location has been introduced as a distributed and decentralized consensus mechanism to localize incorporating agents in a timely manner. PoL reaches consensus when verification regarding the presence of an agent at a certain point is attained in due course. After the agent's location has been broadcasted throughout the blockchain network, other agents can be confident with the received information concerning location coordinates without trusting the broadcasting agent itself. Dynamic proof of location also provides a permissionless and autonomous network of radio beacons that exploits decentralized time synchronization to provide conserved location verification services [86], [121], [122].

28) Proof of Credibility

Proof of credibility has been proposed as an underlying consensus mechanism for the detection and prohibition of invalid news within social networks. Performance evaluation of this approach indicates 89% precision in the detection of fake and tampered news. Proof of credibility considers each user in the social network as a peer that contributes to a distributed ledger that indicates immutable and cryptographically secured logs of discovered rumors. Each block comprises several invalid or tampered news. It is appended to the blockchain network after complying with a pre-defined number of rumors that should be incorporated in each block. Finally, the detection procedure performed by proof of credibility is shared among all peers within the social network platform [87].

29) Proof of History

Proof of history has been introduced to tackle issues associated with intensive computation. This approach executes the SHA-256 hashing algorithm in a consecutive manner to exploit the output of each round as the corresponding input to the subsequent round. Leaders are in charge of confirmation and integration of each transaction with the prevalent hash. PoH is known to be energy efficient as it does not perform intensive mining procedures in comparison with traditional PoW. However, it leans towards wealthier leaders which results in a more centralized and deterministic process and requires more capacity due to consecutive execution of the hashing function [123], [88], [43].

30) Proof of Individuality

Proof of individuality proceeds through individual-to-individual verification where participants are arranged into groups of 5 and each group is in charge of a virtual (video) pseudonym party hangout within the same periods. Participants of each group are required to ensure their counterparts are involved in just one hangout at each interval. Hence, each participant validates POI of the others and, with the termination of hangouts, every contributor is aware of the POI which belongs to each user. However, this consensus mechanism cannot ensure security as physical pseudonym parties and it is difficult to implement due to its reliance on Ethereum, specifically in terms of security [81], [124].

31) Proof of Personhood

Proof of personhood has been proposed as an underlying consensus mechanism that maintains anonymity by exploiting collective ring signatures. Ring signatures do not disclose the keys that have been employed for the computation of a designated signature. This property allows PoP to acknowledge each agent using a cryptographic identity that correlates their physical and virtual identities. PoP allows agents to become miners after being substantiated and configuring a mining pool. Hence, the prospective block proposer will be elected by applying a RandHound method on the ledger [81].

32) SIEVE

SIEVE was initially employed by Hyperledger as an underlying consensus mechanism that tolerates non-determinism. Once performed by distinct replicas, it results in contrasting outputs. SIEVE considers the blockchain as a block box that compares the results from different replicas to sieve out the sequence of diverging outputs. If the diverging values within a procedure reach a certain threshold, the procedure is eliminated [82], [15], [125].

33) Proof of Stake

Proof of stake (PoS), which was initially proposed for Peercoin, has been used as an alternative to PoW to eliminate the excessive power consumption of nodes. Since the election of the block proposer based on the account balance seems unfair, many proposed solutions incorporate the stake size. Although PoS is energy efficient in comparison with PoW, it is not resilient to attacks. Accordingly, several blockchain solutions initially employ PoW and gradually transform to PoS [83], [126].

34) Proof of Elapsed Time

Proof of elapsed time is an underlying consensus mechanism which has been initially proposed by Intel to improve energy efficiency and eliminate the waste of resources. PoET is highly dependent on dedicated hardware to restrict cooperation and decentralization. PoET reaches consensus by the random election of block leaders where the winning odds are spread out evenly throughout the network, and each node holds the same chance of becoming the winner [84].

35) Ripple/Proof of Correctness

Ripple is a consensus mechanism that incorporates validating nodes to preserve a set of trusted nodes acknowledged as Unique Node List (UNL). To append transactions into the ledger, UNL is required to maintain agreement among 80% of the nodes. UNL nodes verify the transactions and broadcast their corresponding votes to the network. Unverified transactions are discarded and retained in the open ledger until meeting the validation criteria. As long as the number of faulty nodes remains under 20%, the ledger is authentic [85], [127], [128].

36) Proof of Energy

Proof of Energy is a consensus mechanism for administration of the P2P energy trading using DLTs. PoE uses smart contracts for regulating energy transactions without excessive energy consumption. After the validation of each smart contract, the next block generator needs to be elected to decide on the next offer. The block generator is elected using the proof of energy that incorporates a consumption-production function to calculate the self-consumption proportion of each prosumer. The user that retains equal consumption and generation is chosen as the block proposer and incentivized accordingly. This approach empowers the prosumers to enhance the operation of both distribution and transmission systems [89].

37) Proof of Witness Presence

Proof of Witness Presence has been introduced as a consensus mechanism for location verification to facilitate situation awareness for crowdsourcing applications in smart cities. It aims to securely raise location awareness without disclosing privacy-sensitive information. Proof of location is the vital core of the proof of witness that enables secure verification of the user's location. Collective measurements are required to ensure the presence of users in particular locations, which is then verified in a private or public network. The notion behind proof of witness presence is to affirm the decisions made in the physical space [90].

38) Proof of Ownership

Proof of ownership has been proposed to ensure a trusted execution environment for participants. This procedure can be employed to certify the integrity and ownership of contracts. The proof is established using a block header and pseudonym. The consensus is met when a proposed block generated by a particular trusted execution environment retains most proofs with unique pseudonyms [56].

39) Bitcoin-NG

Bitcoin-NG is a consensus mechanism that employs BFT and Bitcoin's trust pattern. As blockchain consensus, Bitcoin-NG's performance has been significantly enhanced as it dissociates the leader selection and transaction arrangements. Accordingly, the robustness of the blockchain improves as it reaches more bandwidth. This mechanism increases the permissive latency and bounds it to the network diameter, which actualizes the execution of trustless consensus at a global scale [91], [34].

40) Proof of Asset/Proof of Provenance

Proof of asset was initially employed by DigixDAO, an Ethereum inspired platform that eliminates the middleman from real gold transactions. It has been incorporated to track down the gold during different stages of the transaction using the asset card. The asset card contains the transaction information, such as gold bar's serial number and purchase receipt, that is uploaded into a smart contract. Moreover, the tracking process is performed using private keys, which enhances the security and transparency of DigixDAO [129], [92].

41) Proof of Believability

Proof of believability has been proposed as an extension to the PoS consensus mechanism. PoB incentivizes the participants using untradeable tokens known as Servi. Servi serves as a measure of the believability value associated with each user that indicates positive reviews regarding former behaviors. Higher believability value increases the chances for block creation. PoB employs two types of validators: believable validators that are elected algorithmically, and normal validators that are chosen randomly. In case of fraud behavior detection, the node will be penalized by losing all of its stakes [93], [24].

42) Proof of Trust

Proof of trust is a reliable consensus algorithm proposed for open public service networks such as crowdsourcing. It exploits the participant's trust value and RAFT's leader election mechanism as a reference to determine the transaction validators. Incorporation of trust components and incentive measures within PoT prevents the resource pitfalls associated with traditional PoW. Proof of trust can tackle the deceptive behaviour correlated with crowdsourcing network and the scalability challenges of BFT-based consensus algorithms. Proof of trust enhances fairness and security by reaching consensus through four phases as it distributes participants power into different stages to increase scalability and ensure consistency of the procedure [94].

F. PRIMITIVE COMPLIANT EXTENSION CONSENSUS MECHANISM

1) Fast Paxos

Fast Paxos is proposed as an extension to the classic Paxos, where nodes directly transmit their proposed block to the validator and bypass the leader. This approach narrows down the traditional three-message delay to only two-message delay. Saving one message delay accelerates the learning procedure in the absence of collision and attains any required extent of fault tolerance with the least feasible procedures. In case of collision, only a single message delay will be appended, which is still the least number of required message delays for a common consensus mechanism [130], [141], [142].

2) FaB Paxos

FaB paxos has been proposed as the very first Byzantine approach that reaches consensus within a minimum number of communication arrangements while eliminating the expenses associated with digital signatures. FaB paxos is generalized by dissociating fault tolerance proliferation from speed replication. To enhance the processing speed, FaB paxos relies on gracious execution that, in the presence of authentic leaders, can tolerate process failures. The outstanding difference between the previously discussed fast paxos and FaB paxos is their failure models: fast Paxos tends to fail by a crash while FaB Paxos fails more arbitrarily, [131].

3) Raft

Raft in an underlying consensus algorithm that was initially adapted by Quorum. This algorithm is a simplified extension for the paxos algorithm, which is too theoretical and complicated to be well received. Raft incorporates a state replication model in which all transactions are propagated among all participating nodes. It allows a leader node to generate the next block and eliminates the generation of inessential vacant blocks. To endure f faulty nodes in the blockchain network, raft requires the deployment of at least $2f + 1$ nodes within the network [132].

4) XPaxos

Xpaxos was introduced as a state-machine replication consensus mechanism for the XFT protocol and initially employed by Apache Zookeeper. This consensus is crash fault-tolerant and capable of detecting noncrash faults that result in a contradictory state of the framework. To evaluate the performance of xpaxos, it has been employed on Amazon EC2 data centers. Results indicate the significant performance of this scheme as it outperforms the existing BFT protocols in terms of throughput and latency in the presence of geo-replicated settings. There is a special case of message patterns in xpaxos where, in the case of $t = 1$, it can tolerate a single fault with only two active replicas in each span [133].

5) MPaxos

Multipaxos is another variation of paxos that designates a leader to order the received commands and preserves only

TABLE 6: Primitive-Compliant Extensions

Consensus	Scalability	Finality	Adversary Tolerance	Communication Model	Accessibility	Agreement	Incentives	Centralized	Cost
Fast Paxos [130]	N/A	Probabilistic	N/A	Asynchronous	Permissioned	Vote-based	N/A	N/A	↓
FaB Paxos [131]	N/A	Deterministic	N/A	Synchronous	N/A	Vote-based	N/A	N/A	N/A
Raft [132]	Low	Probabilistic	$<50\%$	Synchronous	Permissioned	N/A	N/A	-	N/A
XPaxos [133]	High	Determinstic	$<=33\%$	Asynchronous	Permissioned	Vote-based	N/A	-	↑
RPaxos [134]	High	Determinstic	N/A	Asychncronous	Permissioned	Vote-based	N/A	-	Moderate
MRPaxos [135]	Very High	Deterministic	N/A	Asynchronous	Permissioned	Vote-based	N/A	-	↑
MPaxos [16]	High	Deterministic	N/A	Asynchronous	Permissioned	Vote-based	N/A	-	↑
M^2Paxos [136]	High	Deterministic	N/A	Asynchronous	Permissioned	Vote-based	N/A	-	↑
EPaxos [137]	Moderate	Deterministic	N/A	Asynchronous	Permissioned	Vote-based	N/A	-	↑
WPaxos [138]	Very High	N/A	N/A	Asynchronous	N/A	N/A	N/A	-	↓
FPaxos [139]	Very High	N/A	N/A	Partially synchronous	N/A	vote-based	N/A	-	↓
CPaxos [140]	N/A	N/A	N/A	Asynchronous	N/A	vote-based	N/A	-	↓

one leader at a time. The single leader at a time principle, preoccupied with the computational resources, breaks down the performance, and prohibits it from scaling alongside the deployment dimensions. Moreover, multipaxos is neither susceptible to locality or complicated command patterns, nor it incorporates dependency relations [16].

6) RPaxos

Ringpaxos is an optimized protocol that has been derived from paxos for clustered environments. Unlike paxos, ring paxos arranges the acceptors in a logically conducted ring and allows coordinators to stabilize the communication amongst acceptors. Ringpaxos preserves the reliability of the classical paxos, which can be employed efficiently to ensure safety under asynchronous circumstances. Similar to paxos, Ringpaxos can ensure safety if several coordinators execute concurrently, but may not be able to ensure liveness [134].

7) MRPaxos

Multi-ring paxos has been introduced to scale the group communication protocols to a large number of nodes. This consensus scheme proceeds by parallel orchestrating a boundless number of ring paxos instances. Multi-ring paxos employs an atomic multicast process that allows multicasting messages to groups of receivers and ensures delivery by evaluating the receivers that convey identical messages. The complexity of multi-ring paxos is due to the deterministic incorporation scheme, which results in dynamic load and deviation among engaged paxos rings. In the two-ring execution of multi-ring paxos in the presence of a single message, two learners are associated with two groups. After *learner2* receives the *m* message, it is not able to deliver it as it needs to ensure the execution order by delivering one from *group2*. Hence, it begins to buffer message *m* until the coordinator of *ringpaxos2* realizes its current rate is below the expected rate and puts forward a skip message order to allow *learner2* deliver message *m* [135].

8) M^2Paxos

M^2Paxos is a variation of paxos that leverages the quorums that compound a great number of nodes to accelerate the decisions. It is a scalable and high-performance implementation of paxos that consists of the coordination phase, accept phase, decision phase, and acquisition phase. This allows the consensus mechanism to determine the sequence of commands with the optimal cost of two communication delays in the advent of dispensable workloads. M^2Paxos has been able to address the shortcomings of other variations of paxos, including the single leader layout of paxos and multipaxos that prevents performance scaling and the performance deficiency of epaxos when the number of nodes overtakes seven [136].

9) EPaxos

Egalitarian paxos is another variation of paxos that can tolerate up to two failures and preserve optimal commit latency. EPaxos can achieve high throughput through uniform load balance and slight performance reduction in the advent of crashed replicas. This consensus has been evaluated while implemented on Amazon EC2. In addition, epaxos incorporates fast quorums to convey nonconflicting commands. However, the linear graph-based scheme that it has employed to present the sequence for the execution of the commands might result in confronting complex dependencies. The investigations indicate that the decentralized and uncoordinated nature of epaxos has resulted in the availability and performance consistency of this approach in the presence of local and wide-area replications [137].

10) Wide Area Network Flexible Paxos

WPaxos has been introduced as an extension to paxos with improved throughput and decreased latency. As a multileader approach for WAN deployment, it allows multiple simultaneous leaders to obtain the ownership of objects from one another. This enables the algorithm to conform to the transformations of access locality and maintain object space partitioning. WPaxos has been evaluated by implementation across 5 AWS zones that resulted the superiority of this algorithm amongst other partitioned and leaderless implementations of paxos. WPaxos implements the FPaxos's flexible quorum approach which allows the existence of multiple simultaneous leaders in the object space that is associated with each leader through object space partitioning [138].

11) FPaxos

As a variation of paxos, fpaxos is fast and capable of implementing flexible quorums. Unlike paxos, it only relies on disjoint sets of participants for affirmation of the proposals, which significantly decreases the latency of the network.

TABLE 7: Proof Compliant Extensions

Consensus	Scalability	Finality	Adversary Tolerance	Communication Model	Accessibility	Agreement	Incentives	Centralized	Cost
IPoS [143]	Moderate	Deterministic	N/A	N/A	Permissioned	N/A	✓	-	N/A
DPoC [144]	Moderate	Probabilistic	N/A	N/A	Permissionless	N/A	N/A	-	↑
DyPoS [145]	High	Deterministic	N/A	Synchronous	Permissioned	N/A	N/A	N/A	↓
PoSBoo [146]	High	N/A	<51%	Synchronous	Permissionless	Vote-based	✓	-	N/A
PoX [147]	High	Deterministic	N/A	N/A	Permissioned	Vote-based	✓	-	N/A
PoUW [148]	High	N/A	N/A	Synchronous	N/A	vote-based	✓	-	↑
FPoS [149]	N/A	Deterministic	<30%	Synchronous	Permissioned	N/A	✓	N/A	N/A
TPoS [150]	High	N/A	N/A	N/A	Permissionless	Vote-based	✓	-	↓
LCPoA [151]	High	Probabilistic	N/A	N/A	Permissionless	N/A	N/A	-	↑
Alt-PoW [152]	Very High	Deterministic	< 33.3%	N/A	Permissionless	Vote-based	✓	-	↑
PoP [153]	High	Immediate	N/A	Asynchronous	Permissionless	Vote-based	N/A	-	↓
Magies PoW [154]	High	Probabilistic	N/A	N/A	Permissionless	Vote-based	✓	N/A	N/A
Magies PoS [155]	High	Probabilistic	N/A	Synchronous	Permissionless	Vote-based	N/A	-	N/A
PoRepl [156]	High	Determiistic	N/A	N/A	Permissionless	N/A	N/A	-	N/A
DPoS [157]	Moderate	Probabilistic	<33.3%	Synchronous	Permissioned	Vote-based	✓	✓	↓
DPoW [158]	High	Probabilistic	N/A	N/A	Permissioneless	Vote-based	N/A	-	↓
TaPoS [159]	High	Probabilistic	N/A	Synchronous	Permissioned	N/A	N/A	N/A	↓
PNPoW [160]	High	Probabilistic	N/A	N/A	Permissionless	N/A	✓	N/A	N/A
PoRetrievability [161]	N/A	Probabilistic	N/A	N/A	Permissionless	Vote-based	N/A	-	↓
Multichain [162]	High	Immediate	N/A	Synchronous	Permissioned	Vote-based	✓	-	↓
POEG [163]	High	N/A	N/A	N/A	Permissioned	Capability-based	✓	-	↓
POEC [163]	High	N/A	N/A	N/A	Permissioned	Capability-based	✓	-	↓
LPoS [164]	High	Probabilistic	N/A	N/A	Permissionless	N/A	✓	-	↑
PoTS [165]	Moderate	Deterministic	N/A	Asynchronous	Permissioneless	Vote-based	N/A	-	↑
PoBT [166]	N/A	N/A	<51%	N/A	N/A	N/A	N/A	-	↑
PoWeight [167]	High	Instant	< 33.3%	Partially synchrnous	Permissionless	capabilirt-based	✓	N/A	↓
Threshold Relay	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
DPoR [168]	High	N/A	N/A	N/A	Permissionless	Vote-based	✓	semi	↓
PoAh [169]	High	N/A	N/A	Synchronous	Permissioned	Capability-based	-	-	↓
Ouroboros [170]	N/A	Deterministic	N/A	Synchronous	Permissioneless	Capability-based	✓	-	N/A
Ouroboros Paros [171]	High	Deterministic	N/A	Partially Synchronous	N/A	Capability-based	✓	N/A	↓
Ouroboros Genesis [172]	N/A	N/A	N/A	Partially Synchronous	N/A	N/A	N/A	-	↓
Ouroboros Crypsinous [173]	N/A	Deterministic	N/A	Partially Synchronous	N/A	N/A	✓	-	↑

By representing the quorum size to developers during the replication phase, fpaxos simply allows them to maintain a customized balance among adversary tolerance and latency. This enables absolute scalability that comes at the price of ineffective fault tolerance [139].

12) Cheap Paxos

Most consensus algorithms with asynchronous communication mode need at least $2f + 1$ processors to be able to tolerate the failure of f processors. Accordingly, cheap paxos represents a dynamic paxos variation that proposes the cooperation of $f + 1$ processors that are actually capable of proceeding within the network and keeping the remaining processors as auxiliaries that provide recovery in the failure of leading processors [140].

G. PROOF COMPLIANT EXTENSION CONSENSUS MECHANISM

1) Interactive Proof of Stake

Interactive proof of stake is able to entail communication in the block generation procedure. IPoS minimizes the number of variables that a single miner can go over, which results in augmenting the network's resistance to grinding attacks. Moreover, this algorithm maintains a static balance resistance against drifting attacks since it does not incorporate timestamps or delays. During block generation, to follow the ticket generation rules, instead of initiating the process with one genesis block, the blockchain starts with one genesis block per participant. Tickets are generated using the seed and hold a distinctive value known to miners. Afterward, each block is evaluated in accordance with the ticket score, and the blockchain that has acquired the highest score is incentivized [143].

2) Delegated Proof of Capacity

Fii is a user-friendly crypto platform implementation that alters the perception of participants about cryptocurrencies. This platform, available for all users regardless of their motives, incorporates a delegated proof of capacity consensus that makes this platform compatible with a wide range of end devices. DPoC works based on a set of precalculated hashes that are registered in the mining pool and FiiPOS, which is incorporated as an extra payment accepting feature, initializes the nonce. This is then be followed by submitting the nonce and the corresponding hashes to the mining pool. Finally, incentives are assigned concerning the number of hashes that have been involved throughout the block generation procedure [174], [175].

3) Deduplicated Dynamic Proof of Storage

Deduplicated Dynamic proof of storage(DeyPoS) is a proof of storage extension that aims to effectively update the records that have been outsourced to the cloud server. It facilitates the integrity verification process for the corresponding users. DeyPoS attains a secure cross user deduplication in multiuser cloud storage system using Homomorphic Authenticated Tree (HAT). This approach simply authorizes clients to acquire the ownership of the data they previously uploaded, without imposing an extra upload procedure. HAT is a binary tree that allows DeyPoS to significantly decrease the communication expenses in both proof of storage and deduplication stages. Each node of HAT is in correlation with a data block, and it does not impose any restriction on the number of data blocks. Moreover, it facilitates integrity verification and dynamic operations. Dynamic Proof of storage is mainly employed on dynamic procedures as it incorporates authenticated structures like merkle tree. Within dynamic procedures, DPos eliminates the requirement of regenerating tags for all existing blocks. Instead, DPos incorporates homomorphic message authentication codes and homomorphic signatures that allow regenerating tags only for the updated blocks during any dynamic procedure that comprises inspection of data integrity [145].

4) Proof of Stake Boo

PoS Boo is an extension to Etheurums PoS casper that has been implemented as SHEILDS's underlying consensus. This approach maintains a progressive quality to enhance the resistance against a wide range of malicious attacks, including the 51% attack in which the fraudster should possess the majority of minted coins and risk them on the execution of an attack. In PoS Boo, the reward will be calculated as a multiplication of the network weight and fixed block reward. This method eliminates transaction censoring of PoW by choosing the block creators randomly and the validators globally. PoS Boo penalizes the miners that contribute to a fork by voting for more than one block with the same height or voting for a false block greater than a preset threshold [176], [24], [146].

5) Proof of Exercise

Proof of exercise is a sustainable extension to PoW that allocates a matrix-based logical challenge acknowledged as an eXercise to each miner in the blockchain network. The procedure initiates by allocating the miner's valid transactions to a random exercise matrix to find the corresponding solution to each matrix. The solutions are published to verifiers for endorsement until they satisfy the minimum number of required validations. Afterward, the entire transaction procedure (including the exercise and verification details) is appended as a new block to the blockchain network [147].

6) Proof of Useful Work

Proof of useful work is an alternative for PoW that operates based on the delegation of low-degree polynomial problems. Within this approach, miners are expected to approach the problems posted by delegators. Contributors can choose among the problems that have been posted to mine the block. After termination of the mining procedure, the miner appends the proof of use details into the block that enables verifiers to inspect whether the problem has been solved using the hash of the block. Proof of useful work can be applied to a wide range of practical problems by preserving their PoW qualities as it is also acknowledged as a delegation for computation methods [148], [24].

7) Fair Proof of Stake

Unfairness is one of the most significant drawbacks of PoS since miners with the greatest share of stake are more likely to generate blocks, which results in collecting more incentives. Moreover, PoS is vulnerable to multibranching attack and this problem needs to be addressed in PoS extensions. Hence, the fair proof of stake has been proposed as a consensus algorithm that incorporates exponential distribution in the selection of block originators, which was formerly performed by uniform distribution. It also successfully reduces the number of forks and their corresponding length that minimizes the attacker's acquisition [149].

8) Trustless Proof of Stake

TPoS has been proposed as an underlying consensus mechanism for Stakenet (XSN) that intends to establish an integrated decentralized structure towards an operative investment implementation. TPoS ensures security of the blockchain by incorporating merchant nodes that can be either the participant itself or the representative hired by participants in exchange for an agreed commission on the participant's desired amount of XSN for staking. The merchant nodes have been authorized to validate the transactions and are not involved in block creation. However, the stakeholders that comply with the minimum collateral requirements can run master nodes that are authorized to vote, verify transactions, and generate blocks as well [177], [178]

9) Limited Confidence Proof of Activity

LCPoA is a consensus algorithm proposed for IZZZIO network. This mechanism leverages proof of activity and limited confidence that imposes a restriction on rewriting blocks. This feature enhances the resistance of LPCoA to 51% attack by confining the target nodes so that the attack can be performed only on a limited number of blocks, which does not affect the network. To track the rewriting block, LCPoA generates automatic checkpoints to reduce the chances of rewriting the blocks [179], [180], [24], [181].

10) Alt-PoW

The adaption of PoW attains consensus in a slow and energy-efficient manner since this approach does not allow participants to gain any perception of progression as it does not provide any information on how far they are from the solution. Accordingly, Alt-PoW suggests that providing the aforementioned information for participants simplifies the miner's decision on whether it is worth devoting subsequent resources to keep solving a certain problem. Hence, Alt-PoW aims to provide progress information by fragmenting a problem into a sequence of problems that requires the problem to be solved in several rounds. This allows miners to evaluate their chances for mining the block as they receive information regarding all miners and their corresponding rounds. This approach allows miners to make a viable decision in terms of identifying the right time to commit to a block or suspend the mining procedure [152].

11) Proof of Proof

Proof of proof consensus provides scalability by allowing a recently developed blockchain also referred to as Security Inheriting (SI) blockchain, to derive safety measures from other Security Providing (SP) blockchains. The inheritance procedure of PoP is independent of SP blockchain miners and does not require permission from SP blockchain or authorization from a centralized network or federated entities. Inheritance does not impose any nontrivial or technological restrictions on SI blockchain that employs this protocol, other than eliminating the interaction of SI non-mining users with SP networks since they are required to contribute to retaining the blockchains native tokens. The SI blockchain is incentivized based on the state that has been published in the SP blockchain network. These publications are further referenced once a substitute fork is proposed to the SI network [182], [153], [24].

12) Magi's Proof of Work

MPOW restrains the blockchain network's hash rate by frequently regulating the incentives based on an attraction-repulsion scheme. In order to hinder mining pools and incorporate low end devices in the mining procedure, MPoW increases the incentives during passive mining intervals to prompt network operations in opposition to the dynamic mining intervals so that incentives are decreased to diminish redundant mining resources. However, this approach makes the network vulnerable to 51% attack as it facilitates the procedure to conquer the network hashing power for an adversary [154], [24].

13) Magi's Proof of Stake

After the implementation of Magi's PoW that maintains feasible solo-mining and cannot establish a correlated reward mechanism concerning hash rate, MPoS was introduced. This approach incorporates the same attraction-repulsion scheme; however, the stake weight is relatively proportional to the stake's time span and number of coins. In MPoS implementation, there is a staking time threshold that restricts the accumulation of offline coin age to one week and specifies that increment in the number of coins in a stake does not necessarily guarantee the augmentation of stake's weight [155], [24].

14) Proof of Replication

Proof of replication is a novel implementation of proof of storage that enables servers to persuade users about the replication of certain data in a dedicated storage. Proof of replication provides an interactive protocol to ensure the storage of unique physical copies, prevent deduplication of several copies, and finally assure verifiers that the challenge/response protocol has been incorporated for the accumulation of each replica [156], [31].

15) Proof of Consensus

Proof of consensus maintains consensus, allowing servers on the network to authenticate transactions. PoC adapts the transaction verification procedure of PoS that does not require mining new cryptocurrencies. However, PoC reaches consensus at 80% validation when employed by Casinocoin, which represents the fact that this approach maintains consensus at an augmented validation rate. Each time a new ledger is generated, it is shared across the network. Once its verified, it is referred to as Last Closed Ledger (LCL) and used as a precise reference for future transactions. Once the network reaches a consensus on a valid transaction, it is followed by the generation of a new LCL. However, using proof of consensus, the validation of transactions is mainly confined by the number of servers [183], [184].

16) Delegated Proof of Stake

DPoS has been proposed as an underlying consensus mechanism that outperforms its counterparts, such as PoW and PoS using a block generation procedure that leads to faster transactions. DPoS reduces energy consumption by incorporating a one vote per share mechanism that enhances the number of process coins. Since stakeholders vote for randomly selected witnesses to preserve consensus, they are incentivized and penalized concerning their generated blocks and accomplishments, respectively. However, DPoS suffers from a lack of decentralization as it incorporates an extensive number of validators to reach consensus. Accordingly, it becomes susceptible to primary attacks such as 51% attack, long-range attack, balanced attack, etc. [185], [48], [186] [187].

17) Delayed Proof of Work

DPoW, which was initially adopted by Komodo, is a promising solution for the double-spending issues. Other than blockchain, several other cryptocurrencies have adapted DPoW with multiple features in common, including insufficient staking power and susceptibility to potential attackers. DPoW enhances the security of ongoing transactions by exploiting the established blockchain with an enhanced hash rate. In the case of 51% attack on Komodo, DPoW appends a security layer and integrates existing notaries to ensure the security of the hashes. It also allows notaries to switch between PoW networks if a significant hashing power is provided [158], [188] [24].

18) Transaction Proof of Stake

TaPoS has been proposed as another extension to PoS in which all nodes are required to participate in the introduced security framework. In order to eliminate the reply attacks on forks, TaPoS requires every transaction to contain the hash of the most recent block's header as a proof of validation. This approach reduces the fraud attempts for the generation of alternative chains since stakeholders are constantly validating the blockchain after each transaction [189].

19) Prime Number Proof of Work

Prime number proof of work was originally adopted by Ppcoin and then Primecoin as a non-hashcash PoW mechanism, also referred to as pure PoW. Unlike bitcoin, the lack of currency within the network is adjusted using Moore's law, and the mint rate is established in accordance with complexity of the hashcash. Hence, the attacker is able to take over the network by manipulating the complexity model as it only requires obtaining under 50% of the network's power. Since network security relies on the accuracy of the complexity estimation, a fixed ratio is appointed for the approximation of difficulty among prime chains [160], [190].

20) Proof of Retrievability

Proof of Retrievability is proposed as an extension to Proof of Capacity. It is incorporated as a distributed cryptographic cloud storage that verifies the integration of stored files without requiring to preserve a copy or retrieve the original file. The verification procedure is performed using authentication data where challenges can be addressed without actually possessing the response value [161].

21) Proof of Energy Generation/ Proof of Energy Consumption

Proof of energy generation/consumption are proposed as extensions to the proof of energy. The difference is that PoEG prioritizes the prosumers that preserve higher generation values with reference to consumption values. Accordingly, these proposers are chosen as validators that verify the energy transactions and append them to the blockchain. They are incentivized by the amount of energy. Proof of energy consumption is used to eliminate the peak hour consumption and facilitate the evaluation procedure of transactions for PoEG [163].

22) Multichain Consensus mechanism

Multichain is an extension to PoW that employs round-robin to choose the validator nodes and attempts to address forking by choosing the longest chain. Multichain grants the administrative dominance to the miners of the genesis block. To preserve the mining diversity in the node election procedure, it relies on the rotations of the round-robin. Accordingly, each node gets the chance to append its proposed block to the chain and broadcast that to the rest of the network [162], [191], [34].

23) Leased Proof of Stake

LPoS is an extension to PoS. It is a promising solution to address the uncertainty issues associated with PoS. It incorporates a leasing option to allow nodes with a lower balance to co-operate in the block verification procedure. This scheme creates a flow within the network in which wealthier nodes can lease their funds to nodes on demand. This flow significantly enhances the chances of inferior nodes in solving the blocks. Accordingly, the acquired rewards are shared with the leaseholders. Nodes that preserve a higher amount of leased balance are more likely to be selected for block creation. This scheme makes the network more decentralized and does not allow the network to be ruled by certain members [164], [192], [193].

24) Proof of TEE-Stake(PoTS)

Proof of TEE Stake is an extension to PoS and aims to address the deficiencies of this protocol in terms of security. To prevent long span attacks caused by Nothing-at-Stake phenomena, PoTS employs Trusted Execution Environments (TEEs) to ensure individuals will generate at least one block per height. TEE imposes all verifiers to sign the blocks for exclusively growing heights to protect the network from nothing at stake and malicious verifiers. It can ensure the security without sacrificing the performance of the network by preventing potential grinding or posterior corruptions using cryptographic schemes [165].

25) Proof of Block and Trade

One of the challenges for implementing business blockchains (BBC) in the context of IoT is their lack of feasibility. Increasing the scalability of the BBC to meet IoT criteria, the consensus mechanism is required to be moderate. Accordingly, PoBT has been proposed to make BBC compatible with IoT applications by decreasing the computation time and improving the storage of IoT nodes. The proposed mechanism proceeds with a two-step mechanism, including trade verification and consensus formation, to improve the performance of the network. As the increments in consensus formation time result in decrements in transaction rates, PoBT reduces the number of cooperating nodes and performs verification just for trade. Hence, the network reaches consensus based on the number of participating nodes, which not only increase the transaction rates but also reduce the required bandwidth [166].

26) Proof of Weight

Proof of weight is an extension to Proof of Stake and has been employed by Algorand. When using PoS, the number of tokens held by each participant determines their chance of discovering the subsequent block. However, the proof of weight allocates weight values to participants in accordance with the asset that each user holds in its account. This consensus mechanism makes network resistance to double-spending attacks as far as at least two-thirds of the overall weighted fraction of participants are truthful. Despite the eminence of proof of weight consensus, it is very difficult to incentivize users of such networks as PoW is not developed for generation of passive revenue streams [167].

27) Threshold Relay

DEFINITY [194] incorporates a threshold relay mechanism as a consensus that includes beacons for leader selection and consists of four layers. The layers provide, respectively, registered client information, distributed random beacons, probabilistic leader ranking protocol, and time-stamping. Beacons assign priority ranks to each node, and the blocks that have been proposed by the nodes with higher ranks are more likely to be authorized. Finally, the block that holds the highest rank is sent throughout the network and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3065880, IEEE Access

nodes are allowed to append the corresponding copies to the blockchain [195], [196].

28) Delegated Proof of Reputation

DPoR is an extension that addresses the deficiencies of Delegated Proof of Stake. This consensus is a semi-decentralized mechanism. Unlike DPoS, it does not rely on staking as a preeminent factor resulting in a more constant coin circulation within the network. DPoR employs the reputation factor as a representation of the node's staked value, resource consumption, and contribution to transactions. Accordingly the voting procedure is performed considering the vote weight of each node, which is defined by the reputation value it holds [168].

29) Proof Authentication

Proof of Authentication (PoAh) has been proposed as an alternative consensus mechanism that preserves a lightweight procedure by withdrawing the hash function from PoW. Every node is involved in the ledger updating procedure where nodes are penalized by losing trust values in case of proving invalid authentication. Like PoW, PoAh performs two forms of authentication, including the authentication of each block concerning its source and incrementing the trust value associated with the corresponding validator. The performance evaluation of proof of authentication has shown that this approach outperforms traditional consensus mechanisms such as PoS, PoW, and PoA in terms of latency, computing procedure, and energy consumption [197], [169].

30) Ouroboros

Ouroboros is a PoS extension that has been initially adopted by Cardano [198]. It is a synchronous and permissionless protocol that operates by dividing chains into epochs. Each epoch is associated with a slot leader selected from qualifying stakeholders. Hence, the chances of becoming a block proposer are proportional to the stake of a node. For adversary tolerance considerations, ouroboros operates on a settlement delay that ensures the security of the ledger when transferred among participants. Participants are also incentivized concerning their honest contribution, evaluated through game theory using the participant's collective interest [170].

31) Ouroboros Paros

Ouroboros Paros is a variation of ouroboros that provides security in the presence of fully adaptive fraudulent. Paros operates in a partially synchronous environment and informs the stakeholders about their leading slots in advance. Like Algornad [167], ouroboros paros incorporates the verifiable random function (VRF) for the generation procedure. The VRF is fed by a private key and a nonce that all participants have agreed upon to generate a random number that determines the slot leader [171].

32) Ouroboros Genesis

Ouroboros genesis is the third variation of ouroboros, geared towards security in a partially synchronous environment. Using a novel chain selection mechanism, it overcomes the deficiencies associated with former ouroboros variations concerning the long-range attack. Having the genesis block information, ouroboros genesis allows individuals to enter protocol execution for robust and dynamic operation. As the performance evaluation of ouroboros genesis suggests, this approach retains dependability of the network against a fully adaptive attacker in the dominance of under half of the stakes and preserves the security [172].

33) Ouroboros Crypsinous

Ouroboros Crypsinous is yet another ouroboros variation. It integrates ouroboros genesis with zerocash [199] to ensure the security of the Proof of Stake against adaptive attacks. Also known as Crypsinous, it incorporates a noninteractive zero-knowledge (NIZKs) proof and key privacy to establish zero-like transactions and retain their autonomy. Accordingly, a cheap key erasure is employed by NIZK for leadership proof to prevent revealing the coin value that was formerly performed by ouroboros genesis. This allows explicit construction of the transaction system and stake shifts since the stake distribution is not communicated throughout the network [173].

H. BFT COMPLIANT

1) Hydrachain Consensus

Hydrachain consensus mechanism is proposed as an extension to the BFT, which is highly dependent on a set of validators that confirm the sequence of transactions with low overhead. Round robin is used to select the proposer of the block from a set of validators in each round, and each round is initiated only after receiving more than 2/3 votes in the former round. Implementing this approach, normal operations maintain low overhead since the proposed blocks are presented with the quorum of signatures on the block of the recent height [221], [222], [223], [127].

2) Modified Federated Byzentine Algorithm

mFBA is implemented on BOScoin [224] as an extension to Federated Byzentine Agreement (FBA) that incorporates Proof of Stake to preserve the governance framework. BOS allows users to freeze coins if, across all nodes, their total amount of frozen assets is within a specific range of the number of coins. The frozen node ensures the security and integrity of the blockchain and it can be used to incentivize the operating nodes. If a node is detected to act maliciously and forging the blockchain, the corresponding frozen coins will be surrounded as a common budget [201], [200].

3) Honey Badger BFT

HB-BFT is introduced as another extension to BFT. Unlike other alternatives, it is not concerned about the synchronicity of underlying network or the timing suppositions. HB-BFT is an asynchronous BFT extension that tackles the network's deficient bandwidth with adequate computation

TABLE 8: BFT-Compliant Extensions

Consensus	**Scalability**	**Finality**	**Adversary Tolerance**	**Communication Model**	**Accessibility**	**Agreement**	**Incentives**	**Centralized**	**Cost**
Hydrachain [200]	High	Instant	N/A	N/A	Permissioned	N/A	N/A	N/A	↓
MBFA [201]	N/A	Deterministic	N/A	Synchronous	Permissioned	Vote-based	N/A	✓	↓
HoneyBadger [202]	N/A	Instant	N/A	Asynchronous	Permissioned	Capability-based	N/A	-	N/A
SUMERAGI [203]	N/A	Deterministic	N/A	N/A	Permissioned	Vote-based	N/A	-	↑
Tendermint [41]	Low	N/A	$<33\%$	PArtially synchronous	Both	Vote-based	N/A	-	↓
Istanbul BFT [204]	N/A	Determinstic	N/A	Partially Synchronous	Permissioned	Vote-based	N/A	-	↓
LFB [205]	N/A	Deterministic	N/A	Synchronous	Permissioned	Vote-based	N/A	N/A	N/A
YAC [206]	High	N/A	N/A	N/A	N/A	Vote-based	N/A	-	↓
FBFT [207]	High	Immediate	$<33\%$	N/A	Permissionless	Vote-based	N/A	N/A	↓
ABFT [208]	Moderate	N/A	N/A	Asynchronous	Permissioned	Vote-based	N/A	-	↓
SBFT [209]	High	Determinstic	N/A	Asynchronous	Permissioned	N/A	N/A	-	N/A
Stellar [210]	High	N/A	$<50\%$	Partially Synchronous	Permissioned	Vote-based	N/A	-	↑
PoQoS [211]	Moderate	Immediate	$<=33\%$	Synchronous	Permissioned	Capability-based	-	-	↓
Elpis [212]	High	Deterministic	N/A	Synchronous	Permission-less	Vote-based	N/A	-	N/A
Zyzzyva [213]	Moderate	Deterministic	N/A	Synchronous	Permission-less	Vote-based	-	-	↓
DBFT [214]	High	Probabilistic	N/A	Synchronous	Permissioned	Vote-based	N/A	✓	N/A
PoPT [215]	High	Deterministic	N/A	Synchronous	Permissioned	Capability-based	✓	-	↓
DBFT [216]	High	Deterministic	N/A	Partially Synchronous	N/A	Capability-based	N/A	-	↓
DBFT [217]	High	N/A	N/A	Asynchronous	N/A	N/A	N/A	-	↓
DBFT [218]	High	N/A	N/A	Asynchronous	Permissioned	Vote-based	N/A	-	↓
BFT-SMaRt [219]	High	Deterministic	N/A	Synchronous	N/A	Vote-based	N/A	N/A	↑
Ouroboros-Bft [220]	N/A	Deterministic	N/A	Synchronous	N/A	Vote-based	N/A	N/A	N/A

assets. Nodes agree on the sequence of transactions that have been formerly stored in their buffers. This approach can be considered a transaction processing scheme based on an asynchronous protocol. Investigation performed by Miller showed more than 20,000 transactions per second for networks with less than 40 nodes using HB-BFT [202].

4) SUMERAGI

SUMERAGI is an underlying consensus algorithm for Hyper ledger Iroha that was inspired by the BChain algorithm [225]. SUMERAGI adapts BFT features in tolerating faulty nodes and exploits a global sequence that considers two sets of nodes in which $2f + 1$ nodes are allocated to the first set, and the other set is composed of the remaining nodes. Hence, to validate transactions, $2f + 1$ signatures are required to reach a consensus on every transaction, which makes only the first set of nodes capable of contributing to the consensus procedure. Subsequently, the transaction is verified by the first set of nodes, and then evaluated by the other set in terms of the authenticity of signatures and contents. Accordingly, the ledger is updated, and the transactions corresponding to the hash are sent through the network [203].

5) Tendermint

Tendermint is a Byzantine algorithm that relies on DLS [226] protocol that communicates through round leaders using a star network topology. Tendermint reaches consensus through three stages. Within the pre-vote stage, right after a new block has been proposed using round-robin, the validators are required to decide whether to broadcast a pre-vote for the aforementioned block. To be a validator, Tendermint requires each node to lock its coins, which will be further used to incentivize or penalize the contributing validators. To proceed from one phase to another (pre-vote, pre-commit ,and commit) the block needs to obtain at least 2/3 of the votes for transmission which in the commit phase corresponds to appendance of the proposed block to the network [41], [227].

6) Istanbul BFT

Istanbul BFT is a replication-based consensus mechanism originally employed by Quorum Chain and incorporates PBFT's 3 stage procedure (PRE-PREPARE, PREPARE, and COMMIT). During each round, every node participates in a random selection of the block proposer through the round-robin. Within the execution pattern of Istanbul BFT, each proposed block needs to acquire $2f + 1$ state messages from the validators in order to be verified from one stage to another. After obtaining $2f + 1$ state messages, the commit message is sent throughout the network to finalize appending the proposed block into the blockchain [204].

7) Leader Free Byzantine

All previously discussed consensus algorithms in the context of blockchain consensus are either deterministic or leader-based algorithms derived from BFT. In the BFT consensus, once a leader transmits a message at a slow pace without activating the corresponding timeout protocol, it results in an unstable performance. Hence, leader-free BFT has been proposed as a deterministic approach that develops an asynchronous consensus from a partially synchronous one. It not only preserves the security of a asynchronous consensus,but also integrates the liveness of the synchronous protocol [205].

8) YAC

YAC has been proposed as a decentralized BFT consensus algorithm to address the deficient message transmission and leader associated with the classical BFT consensus mechanism. This algorithm was initially deployed in Hyperledger Iroha to ensure security and liveness of the underlying transactions by tolerating the utmost f faulty validators among $3f + 1$ participating peers. Experimental results evaluating the performance of YAC in Hyperledger Iroha illustrate eminent scalability of this scheme. However, to decrease peers exposed faults, the vote step delay needs to be modified concerning the number of participating validators [206].

9) Federated Byzantine Fault Tolerance

Federated BFT has been incorporated by Ripple and Stellar as their underlying consensus mechanism. Since both these cryptocurrencies perform work on decree currencies, they require incorporating a highly fault-resistant consensus to ensure higher transaction rates along with scalability. In FBFT, each node generates a unique node list (UNL) indicating committed nodes that will be then used for validating transactions. Receiving at least 80% of the UNL votes is required to ensure the verification of the transaction and, thus, it is appended to the blockchain [210], [228].

10) Hashgraph/ABFT

Hashgraph is an asynchronous BFT that retains consensus by imposing restrictions that indicate only messages from genuine nodes will be transmitted. Moreover, the consensus will be preserved unless the attacker is in control of more than 1/3 of the votes. Hashgraph does not rely on votes as it executes a gossip protocol in which each participant is allowed to reach out to its counterparts and perform synchronization. Hashgraph employs an event-based data structure to store transactions that are accompanied by timestamp associated with the generation time of each event. ABFT makes the network resilient since each event carries not only the transactions but also the hashes of the former events signed to ensure cryptographic security [208], [229].

11) Scalable Byzentine Fault Tolerance

Scalable byzentine fault tolerance is proposed as an extension to PBFT. SBFT allows block generators to validate transactions within 24-hour intervals and reaches consensus concerning the approval from at least $2f + 1$ nodes out of $3f + 1$ cooperating nodes. SBFT enhances the scalability of the network and preserves decentralization by providing replication. Agents perform replication as a pillar that is accountable for consensus. Performance evaluation of SBFT has demonstrated its resilience to the propagation of active replicas [209], [230].

12) Stellar Consensus Protocol

SCP is a global consensus that has been proposed as an extension to FBA. SCP employs the same concept where a quorum is considered as a set of nodes that work towards reaching consensus using quorum slices as a subset to procure agreement. Hence, SCP does not need to trust the whole network and reaches consensus using the validator nodes. However, to obtain a comprehensive consensus, it mainly relies on quorum intersections. SCP is composed of a ballot protocol to verify the agreement of nodes on a quorum and the nomination protocol that feeds the ballot [210], [22].

13) Proof of QoS

PoQ is a consensus mechanism that relies on the quality of service. It considers a network as a collection of subregions, where each region nominates a representative node based on its quality of service. Accordingly, a BFT algorithm is applied to nominees to facilitate the election procedure. Evaluation of PoQ indicates a significant enhancement of throughput while preserving a fair environment for participants [211].

14) Eplis

The motivation behind the advent of Eplis was moving towards an accelerated decision within three communication threads by incorporating the locality of the geo-scale implementations. Not only it can stabilize the geo-replication, but it also ensures the linearizability of multishared proceedings. Elpis associates ownership with individuals and enables dynamic ownership transformations (as opposed to the former procedure of assigning an exclusive leader to all commands inconsiderate of their corresponding fluctuant characteristics). Eplis adapts a linear mechanism that results in 2-fold and 3.5-fold acceleration versus XPaxos and BFT, respectively [212].

15) Zyzzyva

Zyzzyva has been introduced to address the shortcomings of BFT by employing a speculation mechanism that eliminates the execution of the three-phase commit protocol prior to replying to the client's request. Accordingly, the order proposed by the primary server is employed to allow the client to discover potential inconsistencies. Zyzzyva relies on three stages of agreement, view change, and checkpoint. It also consists of a fast, two-way case communication pattern. The fast pattern follows a simple regimen in which the primary node forwards the acquired request to the replicas for an speculative implementation. Once the request has been executed by replica, the corresponding response is returned to the client. The two-way pattern is adapted so that the client obtains coherent responses ranging from $2f+1$ to $3f$ instead of $3f+1$. Since Zyzzyva is a state machine replication (SMR) that operates on $3f + 1$ replicas, it allows a client to collect $2f + 1$ coherent responses within a commit certificate that has been spread out among replicas [213].

16) Delegated Byzentine Fault Tolerance

DBFT has been introduced as an underlying consensus for NEO. It is an extension to PBFT that does not impose any restrictions on the contribution of all nodes for appending new blocks. It follows the same principles as DPoS where a particular set of nodes are in charge of validating the transactions and generating new blocks. DBFT integrates a set of ordinary nodes along with bookkeepers: randomly chosen bookkeepers propose the next block and ordinary nodes vote for them. The new block will be appended after preserving at least 66% agreement among bookkeepers [214], [231].

17) Proof of Previous Transaction

Proof of Previous Transaction has been proposed as a BFT-Compliant consensus mechanism for parallel accounting. It incorporates PBFT for the candidate selection procedure.

Using a consistent hash function, it can provide equity in the computing power of the participants resulting in impartial accounting opportunities and avoiding unbalanced incentive assignments. Parallel accounting allows more than one accountant to administer transactions, which results in a possibility of multiple blocks to be appended to the blockchain simultaneously. PoPT increases the scalability of JCledger using parallel accountants that confront large transactions more efficiently [215].

18) Democratic Byzantine Fault Tolerance

DBFT has been introduced as a deterministic, leaderless consensus mechanism to attenuate the drawbacks of classical leader-based algorithms and improve their resilience. It enables cooperating nodes to agree upon an intrinsically democratic decision in terms of execution of the consensus. DBFT can operate in a partial synchronous manner, provide scalability, and ensure termination of the algorithm even in the presence of a defective coordinator [216].

19) DBFT

DBFT has been proposed as a leader-based BFT extension to address the limitation of formerly proposed consensuses using a novel double-response technique. This allows the replica nodes to reply simultaneously and eliminate the requirements for detection of any instability that may cause a performance decrease during speculative implementation. Performance evaluation of DBFT indicates the superiority of this algorithm among similar BFT solutions such as zyzzyva and PBFT, specifically in the presence of Byzantine faults, by preserving load balance, security, and liveness of the network [217].

20) Diversity of opinion Byzantine Fault Tolerance

DBFT has been implemented on a randomized mesh blockchain (RMBC) to diversify the number of participants resembling PoW while preserving the resilience and reliability of the network. Accordingly, DBFT employs a two-stage consensus procedure. During the first phase, a general BFT is applied, considering a high possibility of the presence of malicious nodes. The second agreement phase is applied to classified verifiers randomly chosen through RMBC. To decrease the probability of collision and ensure the integrity of the network, each transaction is executed only if both consensus agreement stages coincide [218].

21) BFT-SMaRt

BFT-SMaRt is a BFT variant, inspired by PBFT, providing reconfiguration support and modularity. It outperforms the former BFT compliant algorithms, such as PBFT [39] and upright [232], in terms of performance, fault-free execution, and rectifying defective replicas. One of the most important features of BFT-Smart is its evolution over time since its introduction in 2007, based on feedback on the applicability of this mechanism in transaction processing engines and application-level firewalls [219].

22) Ouroboros-BFT

Ouroboros-BFT is a BFT compliant consensus mechanism proposed as an extension to the classic Ouroboros protocol. It is a deterministic protocol that incorporates a predetermined round-robin to broadcast transaction blocks. It shares common characteristics with other BFT variants, such as PBFT, in terms of incorporating passive clients. However, unlike the other variants, Ouroboros-BFT provides instant transaction verification and full network speed transaction processing in the absence of faults [220].

IV. ANALYSIS

As discussed earlier, it is crucial for all networks within a decentralized ledger platform to collectively agree upon the consensus regulations. This ensures the authenticity of the ongoing transactions by validating the contributions made to the blockchain. Since the advent of blockchain, there have been a range of consensus algorithms proposed. Their diversity, in terms of communication model, adversity tolerance, and several other factors, makes them applicable to a variety of scenarios, as discussed in Section III. The classification introduced in this survey allows an effective analysis of the 130 reviewed consensus algorithms within the proposed taxonomy and among the common blockchain applications.

A. CONSENSUS DISTRIBUTION

As expected, most of the examined consensus mechanisms (42%) have been actively used in the context of cryptocurrencies. However, 26% of the examined consensus algorithms are not associated with any particular application domain, as shown in Fig. 7. Notably, in several studies that propose alternatives or extensions to improve some deficiencies of existing consensus approaches, no application domain is specified. This either means that the proposed protocol applies to the same application area as the precedent consensus, or that it can be applied to any domain as long as it satisfies its specific application requirements. As a consequence, Dapps, IoT, and cloud computing are the application domains that incorporate consensus algorithms the most. On the contrary, at the time of writing, only 1% of the reviewed consensuses have been used in the smart grid and localization applications, making them the least common blockchain application domains.

The analysis also reveals the distribution of the reviewed consensus algorithms among the different classes of the proposed classification. As also shown in Fig. 7, 38% of the 130 reviewed consensus algorithms belong to PA (pure alternatives). The next three classes with the largest representation are PCE (proof compliant extensions), BCE (BFT-compliant extensions), and PE (primitive complaint extensions).

Another interesting perspective is provided by analysing the distribution of the proposed consensus classes among the top 10 widely used BC platforms in general (regardless of their application), cryptocurrencies (based on blockchain activity matrix [233]), supply chain BC platforms, and healthcare BC platforms. The results of this analysis are summarized in Table 9 and illustrated in Fig. 8. This illustration

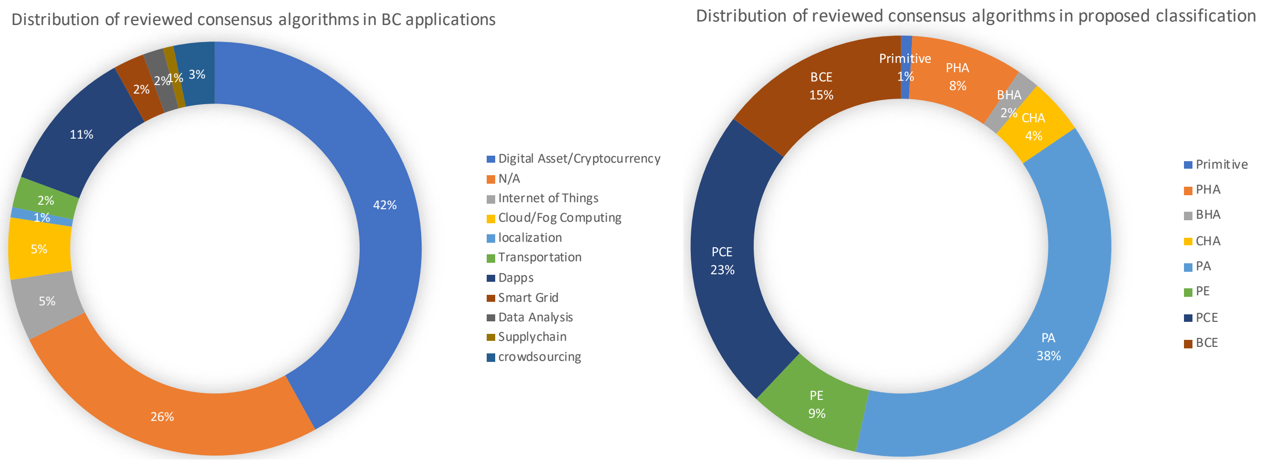


FIGURE 7: Distribution plots

not only shows the distribution of consensus mechanisms among a diverse range of platforms, but also demonstrates how certain classes of algorithms have been designated for the development of specific applications in the context of distributed ledgers. The category that each consensus may fall into is referred to as Primitive, PHA (Proof Compliant Hybrid Alternative), BHA (BFT Compliant Hybrid Alternative), CHA (Cross Compliant Hybrid Alternative), Pure Alternative (PA), PE (Primitive Compliant Extension), PCE (Proof Compliant Extension) and BCE (BFT Compliant Extension).

In this regard, BCE, PCE, and PA are more commonly used to preserve consensus regardless of the application domain. This aligns with the results of consensuses distribution in the proposed classification. In the top 10 blockchain platforms, BCE and PA are used more frequently than other consensuses. PCE have been incorporated in the top cryptocurrency platforms to the same extent as PA. However, PA appears to be the dominant consensus amongst the top 10 supply chain and health-care platforms. In addition, the top 10 supply chain platforms do not employ BCE consensuses. This is an indication of scarcity of this class of consensus among supply chain applications.

B. ANALYSIS OF THE CONSENSUS DISTRIBUTION

The reason behind the dominant adoption of certain classes of consensus mechanisms (e.g. PCE, BCE, PA) in particular application areas can be further viewed from the perspective of the underlying application architecture. Robust consensus and block finality are significant requirements preceding the initiation of communication across chains.

BCE consensus mechanisms employ an exceptional finality that executes high-valued chained transactions very fast and protects blockchain from being forked. BCE algorithms are built within regulatory considerations that make them compatible with business use cases. For instance, health care applications conduct altruistic considerations and interact with genuine identities, unlike exceedingly anonymous and unregulated structures such as Bitcoin. Accordingly, avoiding PCE consensus mechanisms for this type of applications outweighs the risks associated with BCE algorithms. Using BCE algorithms, the consensus decisions are determined conforming to all submitted decisions while eliminating the energy expenditures associated with PCE. Although BCE consensus mechanisms can be applied exclusively to permissioned blockchains due to lack of anonymity, their predominant superiority over other consensus mechanisms is the transaction finality that does not require the confirmation procurement employed by PCE mechanisms.

On the other hand, PCE consensus mechanisms are known to provide significant decentralization by refraining from Application-Specific Integrated Circuits (ASIC). This allows PCE variants to avoid the re-centralization subsequent to decentralization, caused by rising barriers to obtaining mining permissions.

Pure Alternatives are also used where transaction speed and energy consumption are the priorities of the use case. This class of consensus mechanisms can significantly reduce the power consumption and increase the transaction throughput by restraining the computing power of the network. However, the consensus mechanisms of this class are still in their infancy, promising to improve their resilience and mitigate reliance on specialized hardware.

C. EVOLUTION OF CONSENSUS MECHANISMS

With the advent of distributed ledger technology in various application areas, the requirements for consensus protocols significantly raised, especially for protocols that are reliable for both financial institutions and frameworks. This led to the emergence of consensus mechanisms that do not rely on bitcoins' traditional proof of work. Resulting substantial breakthrough brought consensus protocols such as Ripple. It also triggered the emergence of consensus mechanisms that migrated from permission-less systems to token-less permissioned blockchains not allowing anonymous nodes to participate in the verification of transactions.

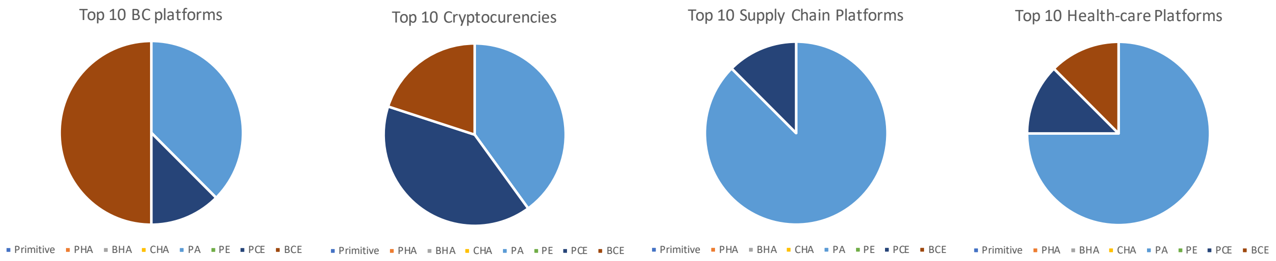


FIGURE 8: Distribution of proposed Classification in BC applications

TABLE 9: Consensus in Different Blockchain Platforms [234]

Crptocurrrency	Consensus	BC Platform	Consensus	Supply Chain BC	Consensus	Health-Care BC	Consensus
EOS [235]	DPOS	Etheurum [236]	PoW	Vechain(VET) [237]	PoA	BurstIQ [238]	PoW
TLOS [239]	DPOS	Hyperledger Fabric [240]	PBFT	Waltonchain(WTC) [241]	Dpos	Factom [106]	Pos Variant
XLM [242]	Stellar	Hyperledger Sawooth [243]	PoET	Ambrosus(AMB) [244]	PoA	Guardtime [245]	PoT
KIN [246]	Stellar	Hedera Hashgraph [247]	ABFT	Modum(MOD) [248]	PoE	Iryo [249]	Pow/PoS
IOST [250]	Proof of Believability	Ripple [251]	RPCA	CargoCoin [252]	PoS	EncrypGen [253]	PoS
ETH [102]	PoW	Quorom [254]	Raft/IBFT	CargoX [255]	PoW	MedRec [256]	PoAuth
BTC [257]	PoW	Hyperledgr Iroha [258]	YAC	ShipChain [259]	PoW	Metadium [260]	PoAuth
BTS [261]	PoW	Corda [262]	ABFT	OriginTrail [263]	PoR	Evernym [264]	Bft Variant
BSV [265]	PoW	EOS [235]	DPOS	TAEL(WABI) [266]	N/A	PeerMountain [267]	No Consensus
TRX [268]	TPOS	Stellar [269]	FBA	Bext360 [270]	N/A	PokitDok [271]	No Consensus

The main concern of any distributed ledger technology is to ensure the security of network transactions. Consensus protocol verifies all transactions have authorized source through agreement on the state of the ledger. Therefore, several consensus protocols have been proposed with different levels of self-enforcing regulations and incentive mechanisms to ensure that participants act legitimately. In addition, the security system of the pioneer consensus protocols such as proof of work does not function effectively for use cases with strict financial regulations. Hence, based on the preferences of an organization at a given time span, different consensus mechanisms can be employed without enforcing a rigid consensus layer.

D. FUTURE PROSPECTS

Since the advent of Hyperledger, the attention has been drawn towards cross-industry and open source distributed ledger solutions improving the cross-compliant hybrid alternative (CHA) solutions introduced in consensus section. Although many providers prefer developing consensus solutions based on specific use case requirements, the need for consensus mechanisms capable of addressing diverse requirements has not yet been fulfilled. Hence, a significant number of consensus mechanisms are expected to emerge in the CHA class.

V. CONCLUSIONS

This paper provides a thorough review on precedent forms of distributed ledger focusing on blockchain and its consensus mechanisms. A total of 130 consensuses have been reviewed analyzing 185 publications ranging from academic journals, industrial websites, conferences, and workshops to technical white papers. To provide a suitable analytical framework, we propose a comprehensive classification of consensus mechanisms based on their building blocks. Accordingly, a consensus mechanism that is not primitive like paxos, is either proof compliant, BFT compliant, primitive compliant, or cross compliant. Consensus mechanisms in the same category tend to share certain characteristics that are discussed in terms of functionality, shortcomings, and advantages.

The proposed classification not only facilitates the analysis of existing consensus mechanisms, but also provides a framework that subsequent algorithms can be related to. Unlike previous attempts for classification of the consensus mechanism, this approach relies on identifying the prevalent features that help to discern the building blocks and communication model of each algorithm. Accordingly, a consensus algorithm may belong to one of the following 8 classes: Primitive, PHA (Proof Compliant Hybrid Altenrative), BHA (Bft Compliant Hybrid Alternative), CHA (Cross Compliant hybrid Alternative),Pure Alternative (PA), PE (Primitive Compliant Extention), PCE (Proof Compliant Extention) and BCE (Bft Compliant Extension).

This article also provides analysis on the distribution of the reviewed algorithms in the proposed categories. As expected, pure alternatives (PA) and proof compliant extensions (PCE) are the dominant categories. Most proposed consensuses are either used in the context of cryptocurrencies or are not associated with any specific application domain. The latter occurs when an alternative or an extension consensus is proposed to improve the deficiencies of their predecessor and its application area is not further discussed. This either means that the consensus is applicable to the same application domain as the precedent consensus or that it can be applied to any domain as long as the requirements are satisfied.

Finally, this survey evaluates the distribution of each proposed consensus group in the top 10 general blockchain, cryptocurrency, supply-chain, and health care platforms. The results indicate that BCE (BFT Compliant Extension), PCE (Proof Compliant Extension), and Pure Alternatives (PA) are most commonly used to reach consensus within a network.

REFERENCES

[1] I. F. T. Alyaseen *et al.*, "Consensus algorithms blockchain: A comparative study," *International Journal on Perceptive and Cognitive Computing*, vol. 5, no. 2, pp. 66–71, 2019.

[2] N. Chaudhry and M. M. Yousaf, "Consensus algorithms in blockchain: comparative analysis, challenges and opportunities," in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*. IEEE, 2018, pp. 54–63.

[3] S. J. Alsunaidi and F. A. Alhaidari, "A survey of consensus algorithms for blockchain technology," in *2019 International Conference on Computer and Information Sciences (ICCIS)*. IEEE, 2019, pp. 1–6.

[4] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, 2020.

[5] 2020. [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/dlt/Pages/default.aspx

[6] 2020. [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/dlt/Documents/d21.pdf

[7] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[8] R. C. Merkle, "A certified digital signature," in *Conference on the Theory and Application of Cryptology*. Springer, 1989, pp. 218–238.

[9] H. Pervez, M. Muneeb, M. U. Irfan, and I. U. Haq, "A comparative analysis of dag-based blockchain architectures," in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*. IEEE, 2018, pp. 27–34.

[10] S. Yang, Z. Chen, L. Cui, M. Xu, Z. Ming, and K. Xu, "Codag: An efficient and compacted dag-based blockchain protocol," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 314–318.

[11] H. Anwar, "Distributed ledger technology: Where technological revolution starts," 2020. [Online]. Available: https://101blockchains.com/distributed-ledger-technology-dlt/#3

[12] N. El Ioini and C. Pahl, "A review of distributed ledger technologies," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2018, pp. 277–288.

[13] 2020. [Online]. Available: https://monax.io/

[14] R. T. Frahat, M. M. Monowar, and S. M. Buhari, "Secure and scalable trust management model for iot p2p network," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019, pp. 1–6.

[15] A. Baliga, "Understanding blockchain consensus models," *Persistent*, vol. 2017, no. 4, pp. 1–14, 2017.

[16] L. Lamport *et al.*, "Paxos made simple," *ACM Sigact News*, vol. 32, no. 4, pp. 18–25, 2001.

[17] 2020. [Online]. Available: https://clamcoin.org/#

[18] 2020. [Online]. Available: https://vericonomy.ams3.cdn.digitaloceanspaces.com/documents/VeriCoin-Proof-of-Stake-Time-Whitepaper.pdf

[19] 2020. [Online]. Available: https://wiki.vericoin.info/index.php?title=Proof-of-Work-Time

[20] J. Benet and N. Greco, "Filecoin: A decentralized storage network," *Protoc. Labs*, 2018.

[21] 2020. [Online]. Available: https://nebulas.io/docs/NebulasTechnicalWhitepaper.pdf

[22] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2017, pp. 1–5.

[23] I. Bentov, R. Pass, and E. Shi, "Snow white: Provably secure proofs of stake." *IACR Cryptology ePrint Archive*, vol. 2016, no. 919, 2016.

[24] A. Shahaab, B. Lidgey, C. Hewage, and I. Khan, "Applicability and appropriateness of distributed ledgers consensus protocols in public and private sectors: A systematic review," *IEEE Access*, vol. 7, pp. 43 622–43 636, 2019.

[25] Z. Liu, S. Tang, S. S. Chow, Z. Liu, and Y. Long, "Fork-free hybrid consensus with flexible proof-of-activity," *Future Generation Computer Systems*, vol. 96, pp. 515–524, 2019.

[26] L. Ren, "Proof of stake velocity: Building the social currency of the digital age," *Self-published white paper*, 2014.

[27] E. Wustrow and B. VanderSloot, "Ddoscoin: cryptocurrency with a malicious proof-of-work," in *10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16)*, 2016.

[28] 2020. [Online]. Available: https://vericoin.info/vericoin-digital-currency/

[29] 2020. [Online]. Available: https://vericoin.info/

[30] T. Moran and I. Orlov, "Simple proofs of space-time and rational proofs of storage," in *Annual International Cryptology Conference*. Springer, 2019, pp. 381–409.

[31] 2020. [Online]. Available: https://filecoin.io/

[32] 2020. [Online]. Available: https://nebulas.io/

[33] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, 2014.

[34] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[35] 2020. [Online]. Available: http://cdn.getlynx.io/2019-03-17_Lynx_Whitepaper_v1.1.pdf

[36] 2020. [Online]. Available: https://getlynx.io/

[37] H. Jennath and S. Asharaf, "Survey on blockchain consensus strategies," in *ICDSMLA 2019*. Springer, 2020, pp. 637–654.

[38] 2020. [Online]. Available: http://docs.icon.foundation/en/whitepaper/_static/LFT.pdf

[39] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[40] S. Duan, S. Peisert, and K. N. Levitt, "hbft: speculative byzantine fault tolerance with minimum cost," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 58–70, 2014.

[41] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, The University of Guelph, 2016.

[42] 2020. [Online]. Available: https://medium.com/@2infiniti/a-primer-to-lft-loop-fault-tolerance-consensus-algorithm-d692bdece85a

[43] L. Ismail and H. Materwala, "A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions," *Symmetry*, vol. 11, no. 10, p. 1198, 2019.

[44] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[45] C. Copeland and H. Zhong, "Tangaroa: a byzantine fault tolerant raft," 2016.

[46] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin meets strong consistency," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, 2016, pp. 1–10.

[47] 2020. [Online]. Available: https://dev-docs.ont.io/#/docs-en/DeveloperGuide/02-VBFT-introduction

[48] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, pp. 118 541–118 555, 2019.

[49] 2020. [Online]. Available: https://credits.com/

[50] 2020. [Online]. Available: https://medium.com/eosio/dpos-bft-pipelined-byzantine-fault-tolerance-8a0634a270ba

[51] B. Podgorelec, V. Keršič, and M. Turkanović, "Analysis of fault tolerance in permissioned blockchain networks," in *2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT)*. IEEE, 2019, pp. 1–6.

[52] 2020. [Online]. Available: https://ontio.github.io/documentation/vbft_intro_en.html

[53] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "Lightchain: A dht-based blockchain for resource constrained environments," *arXiv preprint arXiv:1904.00375*, 2019.

[54] F. P. Junqueira, B. C. Reed, and M. Serafini, "Zab: High-performance broadcast for primary-backup systems," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*. IEEE, 2011, pp. 245–256.

[55] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system, oct. 2008," *URL http://www. bitcoin. org/bitcoin. pdf.(cited on pp. 15 and 87)*, 2017.

[56] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *proceedings of the 1st Workshop on System Software for Trusted Execution*, 2016, pp. 1–6.

[57] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[58] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[59] 2020. [Online]. Available: https://tokens-economy.gitbook.io/consensus/chain-based-proof-of-burn/proof-of-time

[60] G. Ateniese, I. Bonacina, A. Faonio, and N. Galesi, "Proofs of space: When space is of the essence," in *International Conference on Security and Cryptography for Networks*. Springer, 2014, pp. 538–557.

[61] 2020. [Online]. Available: http://docs.proofofexistence.com/#/?id=proof-of-existence

[62] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 2663–2668.

[63] O. Samuel, N. Javaid, M. Awais, Z. Ahmed, M. Imran, and M. Guizani, "A blockchain model for fair data sharing in deregulated smart grids," in *IEEE Global Communications Conference (GLOBCOM 2019)*, 2019.

[64] S. Takahashi, "Proof-of-approval: A distributed consensus protocol for blockchains," Tech. Rep, Tech. Rep., 2018.

[65] M.-L. Marsal-Llacuna and M. Oliver-Riera, "The standards revolution: Who will first put this new kid on the blockchain?" in *2017 ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K)*. IEEE, 2017, pp. 1–7.

[66] M. Swan, "Blockchain thinking: The brain as a decentralized autonomous corporation [commentary]," *IEEE Technology and Society Magazine*, vol. 34, no. 4, pp. 41–52, 2015.

[67] X. Han, Y. Yuan, and F.-Y. Wang, "A fair blockchain based on proof of credit," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 922–931, 2019.

[68] H. Y. Yuen, F. Wu, W. Cai, H. C. Chan, Q. Yan, and V. C. Leung, "Proof-of-play: A novel consensus model for blockchain-based peer-to-peer gaming system," in *Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure*, 2019, pp. 19–28.

[69] 2020. [Online]. Available: https://nexty.io/nexty-whitepaper.pdf

[70] "Flash white paper." [Online]. Available: https://www.flashcoin.io/docs/FLASHWhitepaper.pdf

[71] "The proof-of-cooperation blockchain faircoin," July 2018. [Online]. Available: https://fair-coin.org/sites/default/files/FairCoin2_whitepaper_V1.2.pdf

[72] "Skycoin." [Online]. Available: https://downloads.skycoin.com/whitepapers/Skycoin-Whitepaper-v1.2.pdf

[73] A. Pazaitis, P. De Filippi, and V. Kostakis, "Blockchain and value systems in the sharing economy: The illustrative case of backfeed," *Technological Forecasting and Social Change*, vol. 125, pp. 105–115, 2017.

[74] 2020. [Online]. Available: https://b3coin.io/faq/proof-disintegration-pod-explained/

[75] F. Bravo-Marquez, S. Reeves, and M. Ugarte, "Proof-of-learning: a blockchain consensus mechanism based on machine learning competitions," in *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*. IEEE, 2019, pp. 119–124.

[76] H. Liu, C.-W. Lin, E. Kang, S. Shiraishi, and D. M. Blough, "A byzantine-tolerant distributed consensus algorithm for connected vehicles using proof-of-eligibility," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2019, pp. 225–234.

[77] F. Gai, B. Wang, W. Deng, and W. Peng, "Proof of reputation: A reputation-based consensus protocol for peer-to-peer network," in *International Conference on Database Systems for Advanced Applications*. Springer, 2018, pp. 666–681.

[78] K. Li, H. Li, H. Hou, K. Li, and Y. Chen, "Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain," in *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2017, pp. 466–473.

[79] X. Fu, H. Wang, P. Shi, and H. Mi, "Popf: A consensus algorithm for jcledger," in *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. IEEE, 2018, pp. 204–209.

[80] 2020. [Online]. Available: https://github.com/Bit-Nation/Proof-of-Individuality-POI

[81] M. Borge, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, and B. Ford, "Proof-of-personhood: Redemocratizing permissionless cryptocurrencies," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2017, pp. 23–26.

[82] C. Cachin, S. Schubert, and M. Vukolić, "Non-determinism in byzantine fault-tolerant replication," *arXiv preprint arXiv:1603.07351*, 2016.

[83] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019.

[84] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On security analysis of proof-of-elapsed-time (poet)," in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*. Springer, 2017, pp. 282–297.

[85] D. Schwartz, N. Youngs, A. Britto *et al.*, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, no. 8, 2014.

[86] S. Migliorini, "Enhancing blockchain smart-contracts with proof-of-location," in *10th International Conference on Geographic Information Science*, 2018.

[87] M. Torky, E. Nabil, and W. Said, "Proof of credibility: A blockchain approach for detecting and blocking fake news in social networks," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019.

[88] A. Yakovenko, "Solana: A new architecture for a high performance blockchain," 2018.

[89] P. Siano, G. De Marco, A. Rolán, and V. Loia, "A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3454–3466, 2019.

[90] E. Pournaras, "Proof of witness presence: Blockchain consensus for augmented democracy in smart cities," *arXiv preprint arXiv:1907.00498*, 2019.

[91] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th {USENIX} symposium on networked systems design and implementation ({NSDI} 16)*, 2016, pp. 45–59.

[92] 2020. [Online]. Available: https://github.com/DigixGlobal/digix-press-kit/blob/master/digix-whitepaper.pdf

[93] "Internet of services: The next-generation, secure, highly scalable ecosystem for online services," 2017. [Online]. Available: https://tokeninsight.com/api/upload/whitePaper/InternetofServices_en.pdf

[94] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2018.

[95] 2020. [Online]. Available: https://zookeeper.apache.org/

[96] *Trusted execution environment: what it is, and what it is not*, vol. 1. IEEE, 2015.

[97] 2020. [Online]. Available: https://genesisx.network/GenesisX-WP-v2.0.pdf

[98] K. Karantias, A. Kiayias, and D. Zindros, "Proof-of-burn," in *International Conference on Financial Cryptography and Data Security*, 2019.

[99] 2020. [Online]. Available: https://chronologic.network/

[100] E. Bresson, A. Canteaut, B. Chevallier-Mames, C. Clavier, T. Fuhr, A. Gouget, T. Icart, J.-F. Misarsky, M. Naya-Plasencia, P. Paillier *et al.*, "Shabal, a submission to nist's cryptographic hash algorithm competition," *Submission to NIST*, 2008.

[101] 2020. [Online]. Available: https://coinspace.com/

[102] 2020. [Online]. Available: https://www.chia.net/

[103] 2020. [Online]. Available: https://www.burst-coin.org/

[104] R. Gennaro and M. Robshaw, *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings*. Springer, 2015, vol. 9216.

[105] Y. Zhang, S. Wu, B. Jin, and J. Du, "A blockchain-based process provenance for cloud forensics," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 2470–2473.

[106] 2020. [Online]. Available: https://www.factom.com/

[107] 2020. [Online]. Available: http://lazooz.org/

[108] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1421–1428, 2018.

[109] 2020. [Online]. Available: https://www.poa.network/for-users/whitepaper/poadao-v1/proof-of-authority

[110] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain," in *Italian Conference on Cyber Security (06/02/18)*, 2018.

[111] 2020. [Online]. Available: https://www.nuls.io/

[112] 2020. [Online]. Available: https://www.nuls.io/wp-content/uploads/2019/06/NULS_Whitepaper_2.0.pdf

[113] 2020. [Online]. Available: https://nexty.io/

[114] 2020. [Online]. Available: https://www.flashcoin.io/

[115] 2020. [Online]. Available: https://fair-coin.org/

[116] 2020. [Online]. Available: https://fair-coin.org/sites/default/files/FairCoin2_whitepaper_V1.2.pdf

[117] 2020. [Online]. Available: https://www.skycoin.com/

[118] 2020. [Online]. Available: https://b3coin.io/

[119] 2020. [Online]. Available: https://www.google.com/recaptcha/intro/v3.html

[120] Q. Zhuang, Y. Liu, L. Chen, and Z. Ai, "Proof of reputation: A reputation-based consensus protocol for blockchain based systems," in *Proceedings of the 2019 International Electronics Communication Conference*, 2019, pp. 131–138.

[121] 2020. [Online]. Available: https://foam.space/publicAssets/FOAM_Whitepaper.pdf

[122] 2020. [Online]. Available: https://www.foam.space/

[123] 2020. [Online]. Available: https://solana.com/

[124] 2020. [Online]. Available: https://github.com/proofofindividuality/poi/blob/master/whitepaper.pdf

[125] V. Gramoli, "From blockchain consensus back to byzantine consensus," *Future Generation Computer Systems*, vol. 107, pp. 760–769, 2020.

[126] *Comparative analysis of blockchain consensus algorithms*. IEEE, 2018.

[127] C. Cachin and M. Vukolić, "Blockchain consensus protocols in the wild," *arXiv preprint arXiv:1707.01873*, 2017.

[128] *Ripple: Overview and outlook*. Springer, 2015.

[129] 2020. [Online]. Available: https://digix.global/whitepaper.pdf#/

[130] L. Lamport, "Fast paxos," *Distributed Computing*, vol. 19, no. 2, pp. 79–103, 2006.

[131] J.-P. Martin and L. Alvisi, "Fast byzantine consensus," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 3, pp. 202–215, 2006.

[132] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, 2014, pp. 305–319.

[133] S. Liu, P. Viotti, C. Cachin, V. Quéma, and M. Vukolić, "{XFT}: Practical fault tolerance beyond crashes," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 485–500.

[134] P. J. Marandi, M. Primi, N. Schiper, and F. Pedone, "Ring paxos: A high-throughput atomic broadcast protocol," in *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*. IEEE, 2010, pp. 527–536.

[135] P. J. Marandi, M. Primi, and F. Pedone, "Multi-ring paxos," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*. IEEE, 2012, pp. 1–12.

[136] S. Peluso, A. Turcu, R. Palmieri, G. Losa, and B. Ravindran, "Making fast consensus generally faster," in *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2016, pp. 156–167.

[137] I. Moraru, D. G. Andersen, and M. Kaminsky, "There is more consensus in egalitarian parliaments," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, 2013, pp. 358–372.

[138] A. Ailijiang, A. Charapko, M. Demirbas, and T. Kosar, "Wpaxos: Wide area network flexible consensus," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 1, pp. 211–223, 2019.

[139] H. Howard, D. Malkhi, and A. Spiegelman, "Flexible paxos: Quorum intersection revisited," *arXiv preprint arXiv:1608.06696*, 2016.

[140] L. Lamport and M. Massa, "Cheap paxos," in *International Conference on Dependable Systems and Networks, 2004*. IEEE, 2004, pp. 307–314.

[141] F. Junqueira, Y. Mao, and K. Marzullo, "Classic paxos vs. fast paxos: caveat emptor," *Proceedings of USENIX Hot Topics in System Dependability (HotDep)*, 2007.

[142] 2020. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-112.pdf

[143] A. Chepurnoy, "Interactive proof-of-stake," *arXiv preprint arXiv:1601.00275*, 2016.

[144] 2020. [Online]. Available: https://fiii.io/images/doc/whitepaper.pdf

[145] K. He, J. Chen, R. Du, Q. Wu, G. Xue, and X. Zhang, "Deypos: Deduplicatable dynamic proof of storage for multi-user environments," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3631–3645, 2016.

[146] 2020. [Online]. Available: https://tokens-economy.gitbook.io/consensus/chain-based-proof-of-stake/proof-of-stake-boo-pos-boo

[147] A. Shoker, "Sustainable blockchain through proof of exercise," in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2017, pp. 1–9.

[148] Z. Dong, Y. C. Lee, and A. Y. Zomaya, "Proofware: proof of useful work blockchain consensus protocol for decentralized applications," *arXiv preprint arXiv:1903.09276*, 2019.

[149] A. Begicheva and A. Kofman, "Fair proof of stake," Tech. Rep, Tech. Rep., 2018.

[150] [Online]. Available: StakeNetWhitepaper,Jul.2018,p.55.Accessed:Jul.14,2018.Available:https://stakenet.io/Whitepaper_Stakenet_V3.0_EN.pdf

[151] 2020. [Online]. Available: https://www.researchgate.net/publication/331942292_Applicability_and_Appropriateness_of_Distributed_Ledgers_Consensus_Protocols_in_Public_and_Private_Sectors_A_Systematic_Review

[152] S. Sharkey and H. Tewari, "Alt-pow: an alternative proof-of-work mechanism," in *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*. IEEE, 2019, pp. 11–18.

[153] 2020. [Online]. Available: https://mirror1.veriblock.org/Proof-of-Proof_and_VeriBlock_Blockchain_Protocol_Consensus_Algorithm_and_Economic_Incentivization_v1.0.pdf

[154] J. Lao, "A network-dependent rewarding system: proof-of-mining," *arXiv preprint arXiv:1409.7948*, 2014.

[155] 2020. [Online]. Available: https://www.m-core.org/resources/mining.html

[156] J. Benet, D. Dalrymple, and N. Greco, "Proof of replication," *Protocol Labs, July*, vol. 27, p. 20, 2017.

[157] 2020. [Online]. Available: https://bitshares.org/technology/delegated-proof-of-stake-consensus/

[158] 2020. [Online]. Available: https://docs.komodoplatform.com/whitepaper/introduction.html

[159] "Bitshares history: Transactions as proof-of-stake (tapos) — steemit," 2020. [Online]. Available: https://steemit.com/bitshares/@testz/bitshares-history-transactions-as-proof-of-stake-tapos

[160] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," *July 7th*, vol. 1, no. 6, 2013.

[161] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, 2009, pp. 43–54.

[162] 2020. [Online]. Available: https://www.multichain.com/download/MultiChain-White-Paper.pdf

[163] M. Moniruzzaman, A. Yassine, and R. Benlamri, "Blockchain-based mechanisms for local energy trading in smart grids," in *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT)*. IEEE, 2019, pp. 110–114.

[164] 2020. [Online]. Available: https://docs.wavesprotocol.org/en/blockchain/leasing#leasing-benefits-for-the-node-owner

[165] S. Andreina, J.-M. Bohli, G. O. Karame, W. Li, and G. A. Marson, "Pots-a secure proof of tee-stake for permissionless blockchains." *IACR Cryptology ePrint Archive*, vol. 2018, p. 1135, 2018.

[166] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "Pobt: A light weight consensus algorithm for scalable iot business blockchain," *IEEE Internet of Things Journal*, 2019.

[167] 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*, 2017, pp. 51–68.

[168] T. Do, T. Nguyen, and H. Pham, "Delegated proof of reputation: a novel blockchain consensus," in *Proceedings of the 2019 International Electronics Communication Conference*, 2019, pp. 90–98.

[169] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, "Proof-of-authentication for scalable blockchain in resource-constrained distributed systems," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–5.

[170] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*. Springer, 2017, pp. 357–388.

[171] B. David, P. Gaži, A. Kiayias, and A. Russell, "Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2018, pp. 66–98.

[172] C. Badertscher, P. Gaži, A. Kiayias, A. Russell, and V. Zikas, "Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 913–930.

[173] T. Kerber, A. Kiayias, M. Kohlweiss, and V. Zikas, "Ouroboros crypsinous: Privacy-preserving proof-of-stake," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 157–174.

[174] 2020. [Online]. Available: https://web.archive.org/web/20190402135904/https://www.fiii.io/images/doc/fiiicoin.yellowpaper.v01.pdf

[175] 2020. [Online]. Available: https://sylvester-lee.blogspot.com/2018/07/delegate-proof-of-capacity.html

[176] 2020. [Online]. Available: https://whitepaper.io/document/296/shield-1-whitepaper

[177] 2020. [Online]. Available: https://stakenet.io/

[178] 2020. [Online]. Available: https://stakenet.io/Stakenet_Whitepaper.pdf

[179] 2020. [Online]. Available: https://izzz.io/en/

[180] 2020. [Online]. Available: https://docs.google.com/document/d/11ibTJ2-88r-vo64mLvAgKxsJ2vO3fR9sUre2ZPPrWwA/edit

[181] 2020. [Online]. Available: https://medium.com/@izzzio/lcpoa-universal-as-pow-economical-as-pos-c26f6ba90017

[182] 2020. [Online]. Available: https://www.veriblock.com/

[183] "Casinocoin," Aug 2018. [Online]. Available: https://casinocoin.org/consensus-the-core-of-blockchain-technology/

[184] 2020. [Online]. Available: https://casinocoin.org/

[185] 2020. [Online]. Available: https://how.bitshares.works/en/master/technology/dpos.html

[186] 2020. [Online]. Available: https://bitshares.org/

[187] F. Schuh and D. Larimer, "Bitshares 2.0: general overview," *accessed June-2017.[Online]. Available: http://docs. bitshares. org/downloads/bitshares-general. pdf*, 2017.

[188] 2020. [Online]. Available: https://komodoplatform.com/delayed-proof-of-work/

[189] D. Larimer, "Transactions as proof-of-stake," *Nov-2013*, 2013.

[190] 2020. [Online]. Available: https://primecoin.io/

[191] 2020. [Online]. Available: https://www.multichain.com/

[192] 2020. [Online]. Available: https://wavesprotocol.org/

[193] Q. He, N. Guan, M. Lv, and W. Yi, "On the consensus mechanisms of blockchain/dlt for internet of things," in *2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES)*. IEEE, 2018, pp. 1–10.

[194] 2020. [Online]. Available: https://dfinity.org/pdf-viewer/library/dfinity-consensus.pdf

[195] 2020. [Online]. Available: https://dfinity.org

[196] 2020. [Online]. Available: https://tokens-economy.gitbook.io/consensus/thresholdrelay

[197] D. Puthal and S. P. Mohanty, "Proof of authentication: Iot-friendly blockchains," *IEEE Potentials*, vol. 38, no. 1, pp. 26–29, 2018.

[198] 2020. [Online]. Available: https://www.cardano.org/

[199] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 459–474.

[200] 2020. [Online]. Available: https://github.com/HydraChain/hydrachain

[201] L. Smorgunov, "Blockchain and a problem of procedural justice of public choice," in *International Conference on Digital Transformation and Global Society*. Springer, 2018, pp. 13–23.

[202] 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*, 2016, pp. 31–42.

[203] "Hyperledgerarchitecture." [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf

[204] H. Moniz, "The istanbul bft consensus algorithm," *arXiv preprint arXiv:2002.03613*, 2020.

[205] F. Borran and A. Schiper, "A leader-free byzantine consensus algorithm," in *International Conference on Distributed Computing and Networking*. Springer, 2010, pp. 67–78.

[206] F. Muratov, A. Lebedev, N. Iushkevich, B. Nasrulin, and M. Takemiya, "Yac: Bft consensus algorithm for blockchain," *arXiv preprint arXiv:1809.00554*, 2018.

[207] 2020. [Online]. Available: https://learning.oreilly.com/library/view/foundations-of-blockchain/9781789139396/5d9e1324-88c9-4339-b974-31426ec72835.xhtml

[208] "Home," Dec 2016. [Online]. Available: https://www.swirlds.com/

[209] J. Behl, T. Distler, and R. Kapitza, "Scalable {BFT} for multi-cores: Actor-based decomposition and consensus-oriented parallelization," in *10th Workshop on Hot Topics in System Dependability (HotDep 14)*, 2014.

[210] D. Mazieres, "The stellar consensus protocol: A federated model for internet-level consensus," *Stellar Development Foundation*, vol. 32, 2015.

[211] B. Yu, J. Liu, S. Nepal, J. Yu, and P. Rimba, "Proof-of-qos: Qos based blockchain consensus protocol," *Computers & Security*, p. 101580, 2019.

[212] M. Garg, S. Peluso, B. Arun, and B. Ravindran, "Generalized consensus for practical fault tolerance," in *Proceedings of the 20th International Middleware Conference*, 2019, pp. 55–67.

[213] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative byzantine fault tolerance," *ACM Transactions on Computer Systems (TOCS)*, vol. 27, no. 4, pp. 1–39, 2010.

[214] 2020. [Online]. Available: https://docs.neo.org/docs/en-us/basic/whitepaper.html

[215] F. Xiang, W. Huaimin, and S. Peichang, "Proof of previous transactions (popt): An efficient approach to consensus for jcledger," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[216] T. Crain, V. Gramoli, M. Larrea, and M. Raynal, "Dbft: Efficient leaderless byzantine consensus and its application to blockchains," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2018, pp. 1–8.

[217] J. Zhang, Y. Rong, J. Cao, C. Rong, J. Bian, and W. Wu, "Dbft: A byzantine fault tolerant protocol with graceful performance degradation," in *2019 38th Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2019, pp. 123–12 309.

[218] S. Jeon, I. Doh, and K. Chae, "Rmbc: Randomized mesh blockchain using dbft consensus algorithm," in *2018 International Conference on Information Networking (ICOIN)*. IEEE, 2018, pp. 712–717.

[219] A. Bessani, J. Sousa, and E. E. Alchieri, "State machine replication for the masses with bft-smart," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2014, pp. 355–362.

[220] A. Kiayias and A. Russell, "Ouroboros-bft: A simple byzantine fault tolerant consensus protocol." *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 1049, 2018.

[221] 2020. [Online]. Available: https://github.com/HydraChain/hydrachain

[222] 2020. [Online]. Available: https://www.blockchain-council.org/blockchain/what-is-hyderachain-technology-how-it-works/

[223] B. Expert, "Hydrachain dapp," 2020. [Online]. Available: https://www.blockchainexpert.uk/blog/hydrachain-dapp

[224] 2021. [Online]. Available: https://boscoin.io/boscoin/

[225] S. Duan, H. Meling, S. Peisert, and H. Zhang, "Bchain: Byzantine replication with high throughput and embedded reconfiguration," in *International Conference on Principles of Distributed Systems*. Springer, 2014, pp. 91–106.

[226] C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the presence of partial synchrony," *Journal of the ACM (JACM)*, vol. 35, no. 2, pp. 288–323, 1988.

[227] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. IEEE, 2017, pp. 557–564.

[228] C.-T. Huang, L. Njilla, and T. Gen, "Consensus of whom? a spectrum of blockchain consensus protocols and new directions," in *2019 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2019, pp. 1–8.

[229] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance," *Swirlds, Inc. Technical Report SWIRLDS-TR-2016*, vol. 1, 2016.

[230] Y. Jiang and Z. Lian, "High performance and scalable byzantine fault tolerance," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. IEEE, 2019, pp. 1195–1202.

[231] 2020. [Online]. Available: https://neo.org/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3065880, IEEE Access

[232] A. Clement, M. Kapritsos, S. Lee, Y. Wang, L. Alvisi, M. Dahlin, and T. Riche, “Upright cluster services,” in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, 2009, pp. 277–290.
[233] E. DestBest, “blocktivity,” 2020. [Online]. Available: https://blocktivity.info/
[234] 2020. [Online]. Available: https://blocktivity.info/
[235] 2020. [Online]. Available: https://eos.io/
[236] 2020. [Online]. Available: https://ethereum.org/
[237] 2020. [Online]. Available: https://www.vechain.org/
[238] 2020. [Online]. Available: https://www.burstiq.com/
[239] 2020. [Online]. Available: https://www.telosfoundation.io/
[240] 2020. [Online]. Available: https://www.hyperledger.org/projects/fabric
[241] 2020. [Online]. Available: https://www.waltonchain.org/
[242] 2020. [Online]. Available: https://www.stellar.org/
[243] 2020. [Online]. Available: https://www.hyperledger.org/projects/sawtooth
[244] 2020. [Online]. Available: https://ambrosus.com/#home
[245] 2020. [Online]. Available: https://guardtime.com/
[246] 2020. [Online]. Available: https://www.kin.org/
[247] 2020. [Online]. Available: https://www.hedera.com/
[248] 2020. [Online]. Available: https://modum.io/
[249] 2020. [Online]. Available: https://www.iryo.io/
[250] 2020. [Online]. Available: https://iost.io/
[251] 2020. [Online]. Available: https://ripple.com/
[252] 2020. [Online]. Available: https://thecargocoin.com/
[253] 2020. [Online]. Available: https://encrypgen.com/
[254] 2020. [Online]. Available: https://www.goquorum.com/
[255] 2020. [Online]. Available: https://cargox.io/
[256] 2020. [Online]. Available: https://medrec.media.mit.edu/
[257] 2020. [Online]. Available: https://bitcoin.org/en/
[258] 2020. [Online]. Available: https://www.hyperledger.org/projects/iroha
[259] 2020. [Online]. Available: https://shipchain.io/
[260] 2020. [Online]. Available: https://www.metadium.com/
[261] 2020. [Online]. Available: https://bitcoinsv.io/
[262] 2020. [Online]. Available: https://www.corda.net/
[263] 2020. [Online]. Available: https://origintrail.io/
[264] 2020. [Online]. Available: https://www.evernym.com/
[265] 2020. [Online]. Available: https://bitshares.org/
[266] 2020. [Online]. Available: https://www.taelpay.com/
[267] 2020. [Online]. Available: https://www.peermountain.com/
[268] 2020. [Online]. Available: https://tron.network/
[269] 2020. [Online]. Available: https://www.stellar.org/
[270] 2020. [Online]. Available: https://www.bext360.com/
[271] 2020. [Online]. Available: https://pokitdok.com/who-we-help-health-it/

BAHAREH LASHKARI received her MSc. degree in information technology with computer networks orientation. She is currently pursuing her Ph.D.degree in software engineering and intelligent systems at the University of Alberta. She worked as a Software Engineer at Huawei Technologies Ltd. She is currently working as a Research Assistant on blockchain-enabled smart grids under the supervision of Dr. P. Musilek. Her research interests include blockchain, distributed ledger systems and deep learning.

PETR MUSILEK (Senior Member, IEEE) received the Ing. degree (with great distinction) in electrical engineering, and the Ph.D. degree in cybernetics from the Military Academy in Brno, Czech Republic, in 1991 and 1995, respectively. In 1995, he was appointed the Head of the Computer Applications Group, Institute of Informatics, Military Medical Academy, Hradec Kralove, Czech Republic. From 1997 to 1999, he was a NATO Science Fellow with the Intelligent Systems Research Laboratory, University of Saskatchewan, Canada. In 1999, he joined the Department of Electrical and Computer Engineering, University of Alberta, Canada, where he is currently a Full Professor. Since 2016, he served as a Director of Computer Engineering program and an Associate Chair (Undergraduate). Currently, he is an Associate Chair (Research and Planning). His research interests include artificial intelligence and energy systems. He developed a number of innovative solutions in the areas of renewable energy systems, smart grids, wireless sensor networks, and environmental monitoring and modeling.