KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express xxx (xxxx) xxx

www.elsevier.com/locate/icte

Blockchain consensus mechanisms comparison in fog computing: A systematic review

Yehia Ibrahim Alzoubi[a], Alok Mishra[b,*]

[a] College of Business Administration, American University of the Middle East, Kuwait
[b] Faculty of Engineering, Norwegian University of Science and Technology (NTNU), Norway

Received 17 March 2023; received in revised form 8 December 2023; accepted 19 February 2024
Available online xxxx

Abstract

Numerous consensus mechanisms have been suggested to cater to the specific characteristics of fog computing. To comprehensively understand their unique features, performance, and applications in fog computing, it is crucial to conduct a systematic analysis of these mechanisms. For this study, 79 relevant articles were carefully selected based on predefined criteria. Among these articles, 35 employed work-proof-based consensus mechanisms, 24 utilized voting-based mechanisms, and 22 adopted capability-based mechanisms. Among the 26 identified consensus mechanisms, proof of work remains the most prevalent one. It is important to note that the scope of this paper is limited to the research available in the predominant databases at the time of writing. Future research may expand to include additional databases and more recent literature in this domain.

Keywords: Blockchain; Fog computing; Mechanism; Privacy; Security

1. Introduction

Fog Computing (FC) was introduced to help mitigate several issues of cloud computing such as the limitation of the bandwidth, high latency time, and its centralized structure [1]. FC provides a potential solution for these issues; however, FC seems to have several issues due to its characteristics including its closeness to the IoT devices, homogeneity, distribution among different trust domains, and resource constraints [2,3]. These characteristics created new issues for FC especially security and privacy-related issues [4,5].

Fig. 1 summarizes the characteristics of FC [6–8]. FC exhibits various characteristics that make it a powerful paradigm. Firstly, it supports location awareness and low latency by deploying fog nodes across different locations, bringing processing closer to end devices for reduced delays. Additionally, its geographical distribution sets it apart from the centralized cloud, allowing services and applications to be deployed flexibly anywhere. The scalability of fog computing enables its efficient operation in large-scale sensor networks that monitor the environment. Furthermore, its mobility feature enables direct connections to mobile devices, facilitating smooth mobility methods. In contrast to batch processing in the cloud, fog applications offer real-time interactions between fog nodes, enhancing responsiveness. The heterogeneity of fog nodes and end devices, originating from various manufacturers, is efficiently managed by FC, accommodating diverse platforms. Lastly, fog components exhibit interoperability, enabling seamless collaboration across different domains and service providers.

A large number of articles have been published during the last five years discussing FC's challenges, especially privacy and security drawbacks [9]. Recently, Blockchain (BC) has gained great attention in the literature as the most suitable solution to most FC security and privacy-related issues [10]. Accordingly, a tremendous number of publications have recommended adopting BC in FC, recently [11].

BC can provide many lucrative benefits for FC due to its unique characteristics including [12–14]: (1) decentralization such that the verification of each transaction is done using the BC using a consensus between distributed peers, (2) trustless such that the verification process does not require the peers to trust each other, (3) scalability such that it allows many new participants (participants or peers) to join the chain which

* Corresponding author.
E-mail addresses: yehia.alzoubi@aum.edu.kw (Y.I. Alzoubi), alok.mishra@ntnu.no (A. Mishra).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2024.02.008

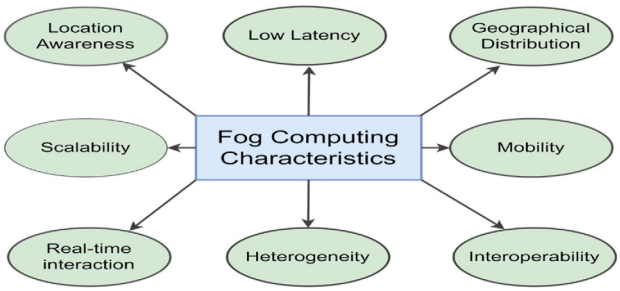

Fig. 1. FC characteristics.

scales up the capacity of the FC, (4) security such that the security of each transaction is achieved using cryptographic techniques that guarantee transparency of the transaction to everyone in the network as well as the guarantee that there will be no one single point of failure, (5) privacy such that no third party will be involved which guarantee the anonymity of data and identification, (6) BC can guarantee the tamper-proof storage of authorized transactions, and (7) accessibility such that BC allows the accessibility to the distributed access control information rules any time and failure of some servers will not prevent access to this information.

Consensus mechanisms are used in BC to achieve agreement among all peers about the verification and validity of a transaction [15]. The first and the most popular consensus is Proof of Work (PoW) [14]. In PoW, nodes are only allowed to release their blocks after performing high effort using their computing power. Despite its high popularity, many studies have reported several drawbacks of PoW including security and privacy issues. Therefore, many alternatives have been since introduced to overcome the drawbacks of PoW [16]. One alternative to PoW was the Proof-of-Stake (PoS) which employs each node stake and a factor to decide the block appending node [17]. In addition to these two mechanisms, many mechanisms depending on the application or context applied, have been introduced recently such as Byzantine Fault Tolerance (BFT), Practical BFT (PBFT), Ripple, and Tendermint [18]. Some studies proposed a modification of the existing mechanisms to suit the context of FC [19]. Nevertheless, implementing these mechanisms or modifications is still not clear. Also, the benefits of the new suggested mechanisms are still confusing, in some cases [20]. Therefore, there is a need to provide a systematic and more comprehensive understanding of BC consensus mechanisms in the FC context [21–23].

To the author's knowledge, the consensus mechanisms of BC-based FC applications have not yet undergone a systematic review [8,24,25]. Moreover, these mechanisms have not been thoroughly examined in terms of decentralization, security, and performance, which are significant aspects when evaluating the service effectiveness of BC consensus mechanisms [15]. The mechanisms have been evaluated subjectively and abruptly in the majority of comparative studies, which makes it difficult to assess the complexity of consensus mechanisms [22]. Furthermore, several consensus mechanisms are not included in the previous comparison since BC consensus mechanisms evolve every day. This paper responds to this demand by conducting a Systematic Literature Review (SLR) of the pertinent literature and offering a thorough analysis of the state-of-the-art consensus mechanisms used in BC-based FC applications. This paper provides a comparative discussion of the features and benefits of these mechanisms and the open questions and future directions of these mechanisms. Hence, this paper aims to answer the following research questions:

RQ1: How are the BC consensus mechanisms used in BC-based fog computing applications?

RQ2: What are the future challenges for BC consensus mechanisms in BC-based fog computing applications?

This paper used the SLR approach to address the research questions. We looked through a variety of databases to find articles that discussed employing consensus mechanisms in the context of BC-based FC. 79 articles were found and chosen to perform the analysis after several search iterations and the application of pre-defined exclusion criteria. The following is how this paper contributes to the corpus of literature.

- Up until August 2022, every work that discussed BC consensus mechanisms in the FC context was identified. There were 26 consensus mechanisms found and discussed, which were classified into three categories voting-based mechanisms, work-proof-based mechanisms, and capability-proof-based mechanisms. This classification was necessary to clarify the purpose for which these mechanisms were utilized. This SLR also identified several novel mechanisms that were released recently [26].
- This classification also makes it easier to examine how these mechanisms and connected apps have evolved in recent years.
- Moreover, this paper compares the three consensus mechanisms that may help choose the most suitable mechanism utilized.
- Additionally, this paper outlined the challenges of all consensus mechanisms, identified in this paper, that may help designers in future development. Work-proof-based mechanisms are typically stated as having high computational requirements, while also having good scalability and security. Capability-proof-based mechanisms, on the other hand, are classified as having medium computational requirements, moderate scalability, and medium security.
- Additionally, voting-based mechanisms are described as having a high communication overhead, medium security, and medium scalability.

The rest of this paper is organized as follows. Section 2 presents the background of BC and consensus mechanisms. Section 3 discusses the research methodology. Section 4 presents the descriptive analysis of the selected papers in this SLR. Section 5 discusses the state-of-the-art of features and benefits of these mechanisms. Section 6 evaluates the performance of these mechanisms. Section 7 discusses the implications and the research and the future research directions. Finally, Section 8 concludes this paper. Table 1 summarizes the abbreviations used in this paper.

Table 1
Abbreviations used in this paper.

Abbreviation	Definition	Abbreviation	Definition
AdPBFT	Algorand Delegated Practical Byzantine Fault Tolerance	PoD	Proof of online Duration
BC	Blockchain	PoDL	Proof of Deep Learning
BFT	Byzantine Fault Tolerance	PoET	Proof of Elapsed Time
DLPBFT	Double-Level Practical Byzantine Fault Tolerance	PoL	Proof of Learning
DPPoW	Designated Prover Proof of Work	PoS	Proof-of-Stake
DPoS	Delegated Proof-of-Stake	PoSer	Proof-of-Service
ePoW	Enhanced Proof of Work	PoT	Proof of Trust
FC	Fog Computing	PoW	Proof of Work
LPoP	Lightweight Proof of Proximity	RAFT	Reliable, Replicated, Redundant, and Fault-Tolerant
mPBFT	Modified Practical Byzantine Fault Tolerance	SG-PBFT	Score Grouping-Practical Byzantine Fault Tolerance
PBFT	Practical Byzantine Fault Tolerance	SLR	Systematic Literature Review
PoA	Proof of Authority	SSC	Stochastic Selective Consensus
PoC	Proof of Creditability	VANET	Vehicular Ad-hoc Networks

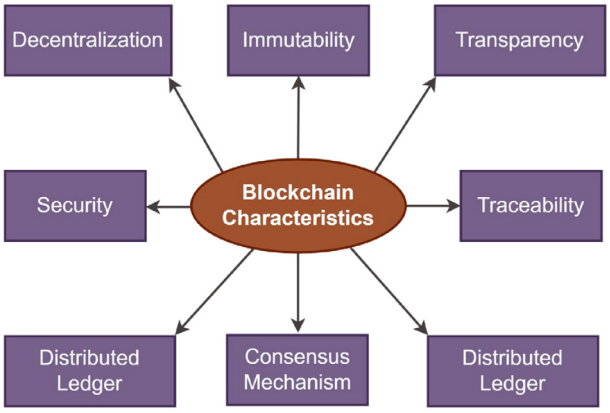

Fig. 2. BC characteristics.

2. Background

2.1. Blockchain characteristics

The BC was proposed in 2008 by Nakamoto [27] to authenticate Bitcoin. BC refers to the shared distributed ledger that manages the growing list of blocks that is secured using cryptography [11]. BC, since then, has achieved high popularity due to its capabilities in maintaining the security and privacy of transactions. In Bitcoin, for example, BC includes two actions that are generated by the participants; the record and the block [12]. Recorded transaction refers to the payment history or personal data, while the block is used to record the transactions. Blocks in BC are appended sequentially to the chain. All nodes or participants are part of the BC community so it has no single point of failure [28].

BC may be divided into three categories in general; permissioned-private (identified participants), permissionless-public (any participants), and permissioned-consortium or federated (can be private or public) [29]. Several BC platforms have been developed that enable the private, public, or consortium BC. Some of the BC platforms are sponsored by companies (e.g., Ethereum which is by the Ethereum Foundation located in Switzerland, and Hyperledger Fabric which is maintained by IBM and the Linux Foundation) and others by a large number of independent developers (e.g., Bitcoin project has a large open-source developer community) [26]. Bitcoin was the first platform that is a public peer-to-peer network of nodes (i.e., participants and general nodes) [30]. Ethereum, which is a consortium BC, has initiated a new epoch of BC since the payments and money are built-in. Also, in Ethereum, financial systems are accessible by everyone (i.e., no one person or company is the controller) and customers own their data [12]. Hyperledger, which is an open-source platform, has had a high impact on BC applications. The most popular project resulting from the Hyperledger platform was the Hyperledger-Fabric which is a private BC [31]. Hyperledger-Fabric has been deployed as a base for other platforms like the IBM platform. General-purpose languages can be used in the development of distributed BC applications [32].

Fig. 2 summarizes the BC characteristics. In a BC system, its fundamental characteristic of "Decentralization" enables multiple nodes to participate in the network, removing the need for a central authority. This decentralization is further supported by the "Consensus Mechanism", where nodes agree on the validity of transactions, preventing double-spending and ensuring consensus. As data is recorded on the BC, "Immutability" comes into play, making it impossible to alter or delete transactions once they are recorded, ensuring data integrity. The principle of "Transparency" complements "Immutability", as all transactions are visible to every participant, promoting trust and accountability within the network. To safeguard data integrity and maintain privacy, BC employs "Security" measures, utilizing cryptography to protect against unauthorized access. The recorded data is distributed across multiple nodes, forming the "Distributed Ledger", providing redundancy and fault tolerance. Additionally, the "Traceability" of transactions, which is made possible through the chronological chain of blocks, allows for a complete audit trail of data history. "Smart Contracts" are another integral feature, self-executing contracts with predefined conditions, automating processes without the need for intermediaries. Lastly, the BC ecosystem can be categorized into "Permissioned" or "Permissionless" based on access control, allowing variations in the level of openness to participants. All these relationships work in harmony to form a robust and transparent blockchain system with applications spanning various industries and use cases.

2.2. Blockchain consensus mechanisms and fog computing

Several industries have been working on the integration of BC and FC applications including IBM, Microsoft, Intel, Cisco, Bosch, Alibaba, and General Electric. The application of BC and FC combination has been introduced in several industrial contexts such as Supply chain management, IoT, Smart grid management, healthcare data sharing, and Vehicular Ad-hoc Networks (VANET). For instance, in supply chain management, FC combined with BC consensus mechanisms to enhance transparency and traceability in supply chains. Through distributed ledgers, stakeholders in the supply chain can securely record and track the movement of goods, ensuring authenticity, reducing fraud, and minimizing delays. In IoT, FC plays a vital role in managing the massive data generated by IoT devices at the edge of the network. BC consensus mechanisms add an extra layer of security, allowing IoT devices to securely communicate and exchange data while preventing unauthorized access and tampering. In healthcare, FC combined with BC consensus to ensure secure and privacy-preserving sharing of medical data among healthcare providers. Patients can control access to their medical records through BC-based permission systems, maintaining confidentiality while allowing authorized parties to access critical information when needed.

The two main perspectives on BC's participation in FC are data processing and communication [19]. In other words, BC will play a critical role in ensuring privacy and security while data is sent from one fog node to another, to the cloud, or to IoT gadgets. For IoT gadgets, the fog node will assume the operator or manager function [2]. The fog node interacts directly with the BC between the cloud and the fog nodes, treating IoT gadgets as consumers [33]. So, the fog node should be stable with the BC functionality to serve its related IoT devices [34]. Thus, permissioned BC platforms such as Hyperledger-Fabric or Ethereum will be the best choice to achieve these goals since it does not face issues of permissionless platforms such as Sybil attacks [1].

Consensus mechanisms in decentralized BC networks establish agreement and trust through a combination of decentralized validation, fault tolerance, cryptographic principles, and economic incentives [32]. These mechanisms enable the creation of transparent, tamper-resistant, and trustworthy distributed systems. Different BC platforms and applications may choose different consensus mechanisms based on their specific requirements and use cases [18].

Every BC-based system makes use of a consensus mechanism in some way. Though there are several ways that the various consensus mechanisms affect the systems. PoW, a pioneering privacy mechanism, is what Bitcoin uses. Originally, Ethereum employed Ethash, which requires a lot of Memory, and then switched to PoS as it requires fewer resources than Ethash. By letting users set up their consensus procedures, Hyperledger adopts a more flexible and open method. Hyperledger offers two distinct mechanisms: BFT and SIEVE, an enhanced variant better suited for commercial applications. A BFT variant and an iterative consensus procedure are both used by Ripple [35]. There is no need to provide an incentive because it is a BC with authorization. A technique similar to PBFT is used by Multichain, although each block only has one verifier, chosen via a round-robin process. Again, this is doable because Multichain is a permissioned BC. Which component is utilized determines the Eris consensus method (e.g., Tendermint uses a variation of BFT). Furthermore, Eris is a private BC where only certain nodes are tasked with verifying transactions [9].

2.3. Blockchain consensus mechanism types

Consensus is a process for ensuring network trust, in which participant nodes in the BC network agree on a new block to be added to the current BC [36]. Consensus mechanisms are used in mining to prohibit the BC from being fabricated, modified, or destroyed [37]. By acquiring and validating the acknowledged BC from the network, nodes might identify the fraudulent transaction inserted by an attacker. As a result, the majority of nodes can agree to cancel the erroneous blocks and which nodes are allowed to attach their suggested blocks. A reliable consensus mechanism is necessary for reaching consensus in a distributed network. The consensus process makes sure that the most recent blocks are correctly added to the BC, the BC data held by the nodes is accurate, and it can even fend off malicious activity [11].

Consensus should generally be taken into account when a network's nodes are malfunctioning or communicating in an erratic manner. Asynchronous or synchronous communication modes should be considered when developing consensus mechanisms [34]. A fork may occur if a new block of changed transactions is added or when valid transactions conflict with invalid ones. Achieving agreement across the participants so that each participant accepts a single valid value is the main objective of the consensus mechanism. Crash failure, Byzantine failure, transitory failure, security failure, omission failure, temporal failure, and software failure are some of the several types of failures reported in the literature [38].

Several BC-based IoT systems have recently attracted interest because of their potential for addressing security problems through smart contract-based verification [38]. To guarantee consistency, smart contracts contain predefined logic that is distributed on a BC network and performed automatically. Smart contracts, on the other hand, are susceptible, which increases the threat for many BC-based systems. There is currently no security mechanism in place to secure smart contracts once they have been deployed [34].

Early on in the BC development, certain BC consensus mechanisms, including PoW, PoS, and PBFT, were implemented. More recently, many novel consensus mechanisms have been proposed. Three categories may be identified for these novel consensus mechanisms [34,38,39]. The first category is the modifications to the initial consensus mechanisms (e.g., Bitcoin-NG, which is an improved PoW, and Algorand, which is an improved PBFT). The second category includes the fusion of the initial consensus mechanisms, such as DBFT, which combines PoS and PBFT. The DAG-based consensus

mechanisms, such as Byteball and Hashgraph, are in the third group. The BC's consensus mechanism can be categorized into two types; the consensus between untrusted nodes and the consensus between trusted nodes. The first type mostly consists of consensus mechanisms ideal for permissioned BCs such as PBFT-based BC and its variant mechanisms, as well as public BCs such as PoS and PoW-based BC. The second type primarily consists of the Paxos, RAFT, and related variant mechanisms. According to many authors (e.g., [16,18,40]), BC consensus mechanisms can be categorized into three types.

1. Work-proof-based mechanisms: The essential idea behind work-proof-based mechanisms is that the node that performs sufficient evidence of computational power will be granted the privilege to add a new block to the chain and get the incentive. Some examples of work-proof-based mechanisms are PoW and DPPoW [21].
2. Capability-based mechanisms: The overconsumption of resources of work-proof-based mechanisms is owing to their competitive nature, in which all participants compete for the opportunity to mine the next block using their processing capacity. As a result, numerous consensus mechanisms have been proposed in the literature to pick a participant based on non-computing capacity including capability-based and voting-based mechanisms. A participant's capability can be determined by a variety of criteria, including the participant's service to the community, the number of cryptocurrencies held by the participant, and the network's trust in the participant. Some examples of capability-based mechanisms are PoS, DPoS, PoET, and PoA [22].
3. Voting-based mechanisms: In voting mechanisms, a participant is elected to generate the BC. This avoids the issue of competition that requires high power consumption of compute-intensive-based mechanisms. It also addresses the issue of the affluent growing richer in capability-based mechanisms. Voting mechanisms are built to withstand Byzantine (i.e., the network's capacity to attain the intended consensus notwithstanding the failure or malicious behavior of some of the system's participants) errors by supposing that the system has distinct participant failures or that some participants are acting maliciously. Some examples of voting-based mechanisms are BFT, PBFT, and DBFT [41].

2.4. Consensus process and Byzantine problem

The classic Byzantine general's problem, in which each general had two options when confronting enemies: retreat or attack, is where consensus mechanisms got their start. They can only reduce fatalities and win a war when all honorable generals concur on the command to retreat or attack. Some of these generals, though, are disloyal and could give the wrong instructions or provide different commands to various generals, undermining the total judgment of the trustworthy generals. In conclusion, the Byzantine general's problem might be defined as the difficulty of convincing trustworthy generals to agree in the presence of untrusted generals [39]. BC is not governed by a single entity and has a distributed ledger. Large rewards might be provided for malicious activities to attempt to cause problems. This makes thinking about the Byzantine problem and its answers important to BC as it affects every component of the system and makes it harder for everyone to reach an agreement on certain decisions [22]. The system is confused by these failures, which makes it challenging for the system to accept the failures. For instance, a server could appear to be down to one participant while still being operational to another. Due to the inability of all participants to agree, the server cannot be declared to have failed [22].

Accordingly, the BC consensus mechanisms are utilized to address the issue of guaranteeing data integrity and availability in the presence of many failed nodes. Byzantine fault nodes and Crash fault nodes are the two categories into which failure nodes may be classified [39]. Byzantine fault nodes exhibit arbitrary behavior, such that, to thwart the process of achieving a consensus, they might transmit incorrect information to other nodes or send alternative information to various nodes. Whereas Crash fault nodes can only stop functioning, such that information will only be dropped or deferred [22]. In the case of a Crash fault, the consensus problem is rather easy to be solved using such consensus mechanisms as RAFT and Paxos. However, in the case of Byzantine fault, multiple participants in a typical BC system frequently represent various entities or consortiums, and because there is no central role, these entities may act randomly. The BC consensus mechanism should thus be able to handle Byzantine fault nodes [39].

In distributed systems, the communication model plays a significant role in the consensus process. In synchronous communication, faults can only occur for a short period. Asynchronous communication, on the other hand, lacks a timeout mechanism. This leads to the fact that, as a result of asynchronous communication, no consensus can be achieved in a distributed system [39]. The bulk of the existing consensus mechanisms relies on either the light synchronous or the full synchronous communication modes, where a timeout method is specified for information delivery. Although information may be deferred in the BC that uses the light synchronization communication mode, it will finally achieve the recipient within a certain window of time after which the transmitting node would be deemed to have failed. Thus, it follows the concept that the availability and integrity criteria of BC consensus methods must be ensured [39].

BC consensus mechanism process may be divided into three stages: representative selection, block addition, and transaction verification [39]. As part of the representative selection stage, the representative node is in charge of creating blocks, which entails gathering and validating transactions, condensing transactions into a block, and delivering the block to other nodes. For each new transaction, the representative may be selected at random, as in the case of PoW or PoS, or it may be selected based on pre-ordering or voting processes such as in PBFT. In the block addition stage, a node verifies the representative and block upon receiving a new block from the representative based on the transaction header. However,

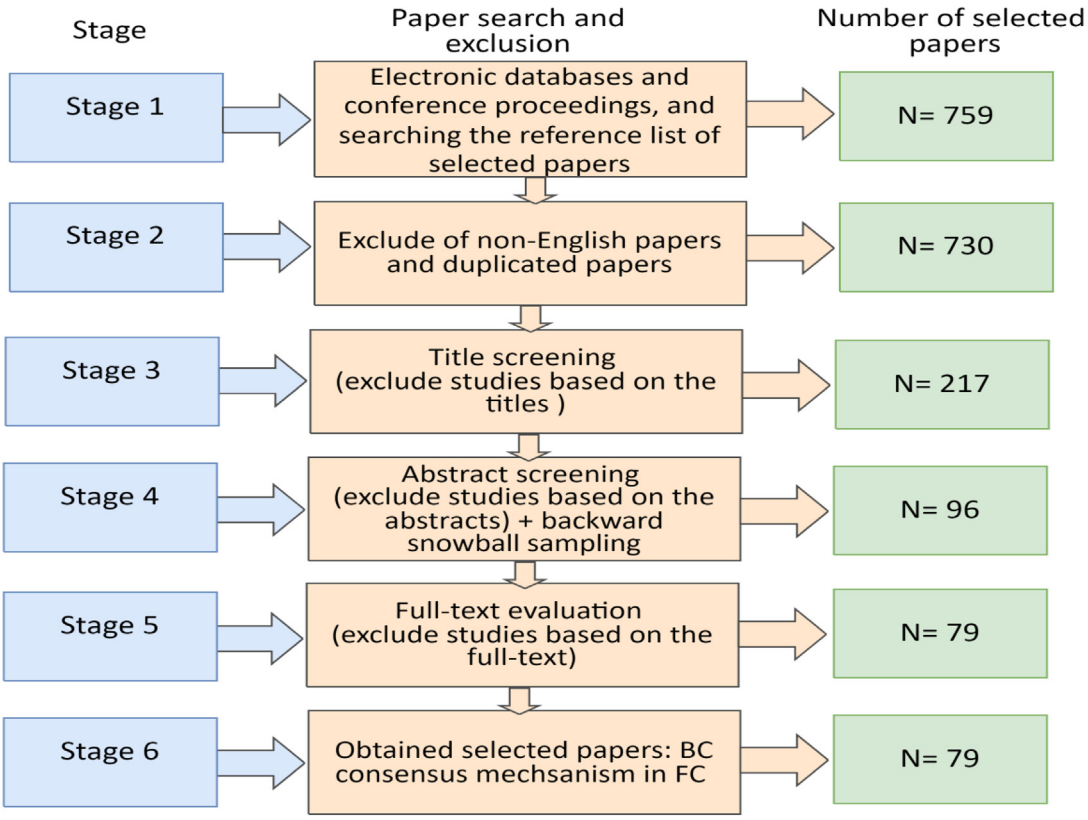

Fig. 3. Study selection process.

the verification of the representative is also related to the method of representative selection. In this stage, the data structure is also validated. After validating the representative node and transaction, the block can be placed in the BC. Some consensus mechanisms like PBFT should achieve the majority of the participant nodes in voting to add, which results in high communication overhead since all participant nodes should share in voting [18]. Each node will have a copy of the BC after choosing the representative node and adding a new block to the BC, which is how the transaction is verified in the third step. The transaction is verified if the verification took place in real-time. Varying nodes may have different BC if there are certain verification delays. Consequently, just because a block appears on the BC does not indicate it has been verified. The BC's data structure affects how a transaction is verified. A block, for instance, can be validated in a PoW-based system if a minimum of six blocks follow it [40].

3. Research methodology

With the help of [42,43] guidelines, we employed an SLR technique to identify the various BC consensus mechanisms deployed for FC. SLR's goal is to locate and synthesize the relevant literature in order to respond to research questions [44]. The SLR protocol is crucial for directing the review process because it offers a framework for comprehending the various BC consensus mechanisms used in FC [44]. In order to validate how the results of this paper were classified, we created a review protocol. Locating studies, selecting and evaluating studies, and extracting and summarizing data are the different phases that have been used.

3.1. Locating studies

This review made use of the nine well-known electronic databases listed below. These databases should provide sufficient literature coverage for this paper. Stage 1 of the exclusion criteria (Fig. 3) was used in this case.

- IEEE Xplore (www.ieeexplore.ieee.org/Xplore/).
- Elsevier ScienceDirect (www.sceincedirect.com/).
- MDPI Online (https://www.mdpi.com/journal).
- Google Scholar (http://scholar.google.com.au/).
- SpringerLink (www.springerlink.com/).
- Wiley Online Library (https://onlinelibrary.wiley.com/).
- Emerald Insight (https://www.emerald.com/insight/).
- SAGE Publication (https://us.sagepub.com/en-us/nam/home).
- ACM Digital Library (www.portal.acm.org/dl.cfm).

In this phase, all possible combinations of BC, FC, and consensus were searched using the logic operators "AND" and "OR". We search the combination of "Blockchain" AND ("fog computing" OR "edge computing") AND "consensus mechanism" OR "consensus protocol" OR "consensus method" OR "consensus algorithm". The articles that were selected include a range of FC applications, including those for vehicles, drones, healthcare products, and smart cities. Peer-reviewed

Table 2
Publication channel.

Database	Study	Number	Percentage (%)	Total percentage (%)
IEEE Xplore (Journal)	[45–88]	44	55.7	60.8
IEEE Xplore (Conference Proceeding)	[89–92]	4	5.1	
Elsevier Science Direct (Journal)	[41,93–106]	15	19	19
MDPI (Journal)	[107–111]	5	6.35	6.35
Wiley Online Library (Journal)	[112–115]	4	5.1	5.1
SpringerLink (Journal)	[116,117]	2	2.5	3.75
SpringerLink (Book Section)	[118]	1	1.25	
Google Scholar (Journal)	[119–121]	3	3.75	3.75
ACM Digital Library (Journal)	[122]	1	1.25	1.25

articles that have been published in journals, conference proceedings, or book sections are the type of articles that were selected in this SLR. Fig. 3 demonstrates the exclusion criteria employed and the count of articles included at each stage in this SLR. The review encompassed research conducted until August 2022. Articles discussing prefaces, poster sessions, article summaries, news, editorial debates, or reader's letters were excluded from this SLR. Only articles written in the English language were considered.

3.2. Study selection and evaluation

The exclusion criteria ranging from stage 2 to stage 5 are discussed in this section. Based on the criteria reported in Section 3.1, the authors independently evaluated all of the literature. At the end of each stage, all authors got down together and discussed the articles that were included and excluded. We kept the chosen studies in EndNote as a citation management tool. The aforementioned search criteria returned 759 hits in total. The number was reduced to 730 after excluding articles published in languages other than English and duplicate studies. Additionally, the number was reduced to 217 once the article titles were reviewed, in stage 3. In this stage, articles that were unrelated to the BC consensus mechanism in FC were eliminated. Some titles, however, were not correctly detected and were thus added to the following review stage.

At stage 4, articles were disqualified based only on their abstracts. The article was rejected if the abstract did not specifically address this SLR scope. The article was left till stage 5 if the abstract was not related to the scope of this SLR. 131 articles were eliminated in this stage, leaving 86 articles for stage 5. However, this stage also applied the backward snowball sampling approach to finding other related articles, which resulted in another 15 articles. These 15 new articles were investigated against the exclusion stages 2 and 3. 5 articles were excluded, which resulted in another 10 articles being included in this stage, bringing the overall stage 4 total article number to 96 articles. In stage 5, the full-text review was applied to every potential article. 20 articles were eliminated from this stage because they failed to provide the BC consensus processes in FC, leaving 79 articles for stage 6 analysis.

3.3. Data extraction and synthesis

The final selection decision was to include any article that discussed using the BC consensus mechanism in FC or edge computing. Therefore, the article was disregarded if it did not address the BC consensus method or had a focus other than on FC or edge computing. Some studies have not identified which consensus mechanism is used in their design such as [1,10,17,19], and [36]. Therefore, these studies were excluded from this review. Other studies reported that some consensus mechanisms can be used, but did not clearly state which mechanism, such as [9,11,37]. Other studies have not used any consensus mechanisms such as [123] (used SDN cluster technique to achieve consensus) and eliminated the usage of Pow in order to decrease resource consumption. Also, [124] suggested utilizing BC and SDN to create clusters in IoT networks. With this method, the energy-intensive consensus mechanism is no longer necessary. Moreover, review articles were excluded from this SLR. The three BC consensus mechanism types (themes) listed in Section 2.3 were adopted. Independently, the authors conducted theme analyses of the articles that were selected. The final comparison of the three themes across all authors yielded a consensus of almost 72%. The 79 items chosen in the theme analysis were all agreed upon by all authors. The next sections provide an analysis of the chosen studies [39–115,123,124], depicted in Table 2.

4. Descriptive analysis

The publication channel for the chosen article is displayed in Table 2. With 60.8% of the total number of selected articles (55.7% journal articles and 5.1% conference proceedings articles), IEEE's IEEE Xplore database received the highest rating. Elsevier Science Direct, the second-most-cited database, had a score of 19% (all journal articles). MDPI Database is in third place with 6.35%, followed by Wiley Online Library with 5.1%, and the ACM Digital Library, which scored the least, with only one article (i.e., 1.25%). Moreover, using the search criteria we used, we were unable to find any related articles in the Emerald Insight or SAGE Publication databases.

Fig. 4 presents the BC consensus mechanisms found in FC through this SLR. In this diagram, we categorize PoW and its extensions (ePow and DPPoW) as PoW-based mechanisms, while PoS and its extensions (DPoS, Improved DPoS, and PPoS) are referred to as PoS-based mechanisms. PBFT and its extensions (BFT, DLPBFT, mPBFT, ADBFT, SG-PBFT,

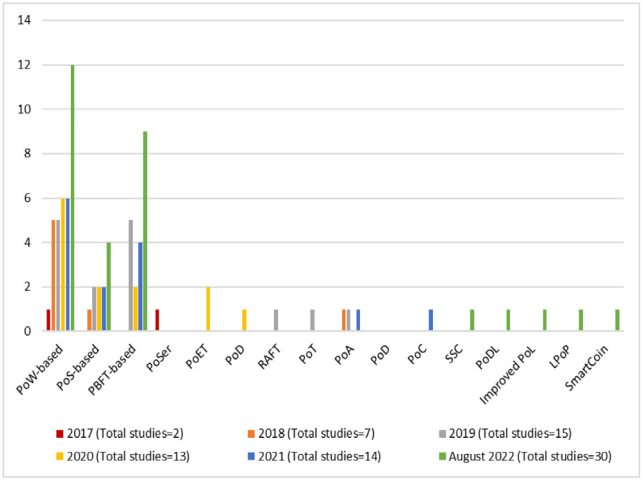

Fig. 4. Consensus mechanism number and year.

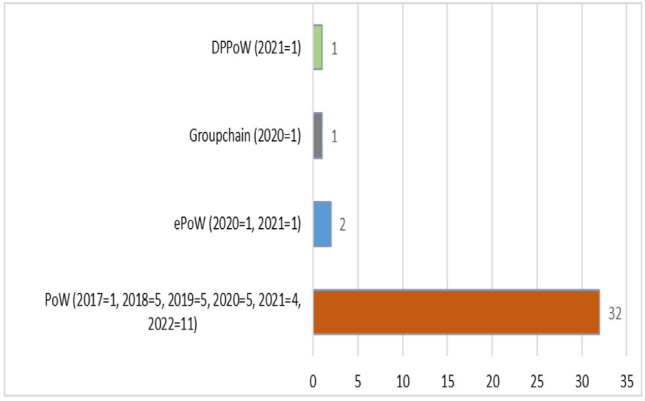

Fig. 5. Work-proof-based consensus mechanisms number.

Simplified PBFT) are grouped as PBFT-based mechanisms to facilitate comparison. However, in the following sections, all these extensions will be individually and thoroughly discussed. The distribution of publications shows that only 2 studies were published in 2017, followed by 7 in 2018, and then 15 in 2019, reflecting the relatively new nature of FC and BC technologies. Until August 2022, a total of 30 studies were reported, as depicted in Fig. 4. Notably, PBFT-based mechanisms witnessed a significant increase in 2022, while PoW mechanisms and their variants have been the most prevalent since 2017. Additionally, 2022 saw the introduction of several other mechanisms, including TPM, PoDL, PoL, and LPoP.

5. Blockchain consensus mechanisms in fog computing

The RQ1 (How are the BC consensus mechanisms used in BC-based FC applications?) will be addressed in this section. Work-proof-based, capability-based, and voting-based mechanisms make up the three categories of BC consensus mechanisms. It is essential to mention that other consensus mechanisms or new extensions have been suggested to overcome the drawbacks of applications that rely on PoW, PoS, and PBFT.

5.1. Work proof-based mechanisms

The implementation of the representative defines phase is a primary concern for consensus mechanisms of the work proof type. This category of mechanisms includes PoW and its variations. Numerous subsequent mechanisms have been influenced by the initial PoW mechanism. In this section, we outline the various PoW-based consensus mechanisms utilized in the BC-based FC systems. Table 3 summarizes the work-proof-based consensus mechanisms identified in this SLR; their characteristics, the purposes of the consensus mechanisms, and the FC applications.

Fig. 5 displays the distribution and count of PoW and its variant mechanisms. Despite facing criticism for its high resource demands, PoW remains the most popular consensus mechanism. In the years 2020 and 2021, only two studies used ePoW, while one article utilized DPPoW in 2021, and another article employed Groupchain in 2020. The figure also illustrates that approximately 5 studies per year focused on PoW deployment between 2018 and 2021. However, in 2022, the number significantly increased to 11 studies. This could be attributed to the way PoW-based applications or solutions are designed. Despite the substantial resource requirements, many of these solutions adopt PoW to enhance transaction security and privacy

5.1.1. Proof-of-work (PoW)

The fundamental idea behind this mechanism is to demonstrate its validity using computational resources. A nonce value is also present in the header of blocks produced by PoW. When preparing a new block, the participant must try various nonce values repeatedly until the block header's hash value meets the challenging criterion, for instance, with several leading zeros [125]. BC validators in a PoW network should accept data from a block header as input and pass it through a cryptographic hash repeatedly. "Mining" is another name for this technique [39]. Every time the input data is processed through the cryptographic hash, the validator includes an anomaly score called a nonce.

When a participant correctly mines a block and the mined block is successfully added to the BC, the participant node winner (leader) earns the mining incentive [56]. Participants who must show that they performed a certain amount of work must do multiple calculations before the solution is provided, which reduces the risk of attacks. This way ensures that BC systems are secure and reliable. PoW is based on the idea that a node that contributes a lot to the network would be less likely to target it [120]. Attackers need 51 percent of the network's computer power to manipulate the BC. This is a huge expense for the attackers, and it is unlikely that they will be able to afford it in practice. This lowers the Sybil attack since participants must perform costly tasks that are virtually difficult to complete by a single participant [75]. However, due to the high computing process and energy usage, transaction throughput is low, which impacts the PoW's scalability capability, which is undesirable in an FC setting [126].

Table 3
Work-proof-based consensus mechanisms.

Mechanism	PoW characteristics	Study	Purpose of consensus mechanism	Application
PoW Wu 20 used smart contracts	• Minimize the attacks up to 50% or less • Improve security (e.g., DoS attack, double spending attack)	[49,51,54, 70,73,86, 89]	Consensus for secure data transmission	• FC resource management [70] • Vehicle FC management [49,86] • Electrical vehicle cloud/FC data security [51,73,89] • FC/IoT data security [54]
	• First and most common mechanism	[52,58,81, 84,88,119]	Consensus for resource management in FC	• FC resource management [52,58] • VANET resource management [119] • Edge computing market [81] • Mobile IoT resource competition [84,88]
		[47,55]	Consensus for cooperative FC services	• Trustworthy FC services [55] • Cloud/FC security [47]
		[93,99,100, 110,113, 120]	Consensus for new block validation	• Cloud/FC security [120] • FC/IoT authentication [93] • Data security [113] • Edge/IoT task offloading [110] • Industrial IoT applications [99,100] • FC/drone data management [98]
		[53,109, 118]	Consensus for unsupervised machine learning	• FC data privacy [53,109,118]
		[56,57]	Enhancing incentivization in IoT applications	• Edge computing resource management [56,57]
		[46,104]	Choosing winner node to add new blocks to the BC	• Cloud/FC service provider competition [46] • 5G network data security [104]
DPPoW	• Distinguish if the string is created by the prover miner	[59]	Consensus for new block validation	• FC/IoT key management
ePoW	• Allow for immutable data sharing and guard against data poisoning threats	[82,96]	Consensus for new block validation	• FC/IoT DDoS attack prevention [96] • Industrial health data security [82]
Groupchain	• Groupchain	[50]	Solving the consistency problem in Bitcoin and bitcoin-NG	• PoW used for leader selection and transactions sequencing
Mixed	• Mixed mechanisms were used to extract the strengths of each mechanism	[48]	PoW was used for interacting among nodes and PoC was used to identify DoS and other attacks	• Cloud/FC task management
		[74]	PoW or PoS were used to verify transactions and DPoS was used to select the winner node	• Edge/IoT resource allocation
		[59]	DPPoW was used for the resource authentication mechanism and PoS was used to select the winner node	• FC/IoT key management

Consensus for secure data transmission: To achieve consensus for secure data transmission, Lei et al. [49] suggested a key management scheme for vehicle FC. The PoW was utilized to accomplish message integrity and security by verifying the signature to ensure node authentication [49]. In electric vehicle cloud/FC, Liu et al. [51], Gu et al. [89], and Kang et al. [73] presented a BC-based scheme, in which the PoW was used to achieve consensus by using the value of energy contribution and the frequency of data contribution. Similarly, Qureshi et al. [86] suggested a strategy using Ethereum BC to enhance privacy preservation in the vehicle system. The PoW consensus was utilized to verify transactions [86]. Sharma et al. [54] and Fan et al. [70] suggested leveraging BC-based distributed mobility management to implement a safe handover method for IoT devices in FC. PoW was used to handle data sharing in peer-to-peer format, which preserves the data in blocks following full control and data validity.

Consensus for resource management: To achieve consensus for resource management in FC, Loung et al. [52] and Yang et al. [58] proposed the construction of an ideal auction based

on deep learning, where PoW was used to achieve consensus when participants purchase one or more computational resource components at the fog nodes. Similarly, for vehicle FC, Kong et al. [119] deployed a PoW consensus mechanism to provide a method for vehicle networking resource management based on resource transactions. The PoW mechanism was used to identify which nodes are eligible to add new blocks to the BC and get associated rewards [119]. Guo et al. [81] developed an architecture for the edge computing market in which several edge service providers can offer computational resources. PoW was used to establish BC's resource-sharing and incentive system [81]. Liang et al. [84] proposed a mining framework for the mobile IoT devices resource competition to reduce the computationally heavy mining loads on participants and to support IoT device involvement. Similarly, Yang et al. [88] used the Stackelberg game theory for resource pricing for mobile IoT devices to maximize the advantages for both the edge server and device users. PoW was deployed to purchase computational resources [88].

Consensus for cooperative FC services: To achieve consensus for cooperative FC services, Kumar et al. [47] deployed PoW mechanisms for cloud/FC transaction sharing; however, the author has used the polynomial matrix factorization and expectation maximization mechanism to reduce the number of iterations to reduce the computation requirements of PoW [47]. Wu et al. [55] introduced the BlockEdge, a BC-based platform that offers trustworthy cooperative FC services, to solve the challenges of trustworthiness between mining nodes. To increase the efficiency of the FC services and produce a puzzle that represents dispersed stakeholder work, PoW was implemented [55].

Consensus for new block validation: To achieve a consensus to validate new blocks, Shukla et al. [93] utilized the PoW-based Ethereum platform to propose a three-tier BC-based FC architecture, where PoW was used to increase the degree of authentication in FC/IoT healthcare applications. In the resource allocation scheme for edge AI computing that Qiu et al. [113] presented, PoW was used to validate the new transaction. A cloud/FC scheme based on BC technology was presented by Nadeem et al. [120]. To protect the privacy of the drivers in a cognitive radio VANET, BC was created as an alternative to the traditional cloud/FC architecture. This scheme integrates the partial contributions from each service provider utilizing PoW, which provides service transparency [120]. Wadhwa et al. [110] proposed a consensus solution based on PoW, in which a sole participant node is chosen for mining the workload, in order to offer an energy-efficient consensus mechanism in edge/IoT computing architecture. The participant is chosen based on the digitalization of the relevant machine's parameters [110]. Similarly, Lakhan et al. [99] suggested a BC-enabled federated learning paradigm for industrial IoT applications to reduce energy usage and application latency. To guarantee the transaction's validity, PoW was utilized. Furthermore, Mohapatra et al. [100] developed a BC-based strategy for fog-based IoT networks to improve data security. Block addition was performed by an authenticated IoT device via PoW [100]. Khan et al. [98] suggested a solution to improve data management in a fog-based drone system employing Hyperledger-Fabric BC, smart contracts, and a metaheuristic mechanism. To maintain preservation, add a new recording to storage, and perform drone enrollment, the PoW mechanism was utilized.

Consensus for unsupervised machine learning: To enhance privacy in unsupervised machine learning, Mahmood and Jusas [109] proposed a federated learning-based BC-based model. By utilizing Ethereum's incentive mechanism and PoW consensus mechanism to foster trust across the decentralized nodes, this model makes federated learning possible [109]. With the use of BC-based federated learning in FC, Qu et al. [53] presented a decentralized privacy-preserving scheme. With the help of a PoW-based BC, this scheme provides unsupervised machine learning to cooperate without the need for a central authority [118].

Enhancing incentivization in IoT applications: To enhance incentivization in IoT applications, to enable IoT devices that employ mobile BC apps, Xiong et al. [57] presented an economical strategy for managing edge computing resources. Participants adopted PoW as the fundamental consensus mechanism to benefit from the incentives [56].

Choosing winner node to add new blocks to the BC: For computing resource allocation in cloud/FC service providers, Jiao et al. [46] presented an auction-based market approach. PoW was employed in order to achieve the consensus on the winner node to add new blocks. Moreover, Wan et al. [104] suggested and combined Wasserstein's generative adversarial network with BC-enabled federated learning to address the challenges of the issues related to centralized cloud/edge architecture. This was done to provide differential privacy so that edge devices' model parameters in 5G or later networks might be protected. To obtain incentives and validate local model changes, mining nodes utilize the PoW mechanism to compete with one another [104].

5.1.2. Designated prover PoW (DPPoW)

The DPPoW is a recently developed PoW extension [59]. DPPoW is based on PoW (i.e., a fog device will be assessed if it has enough computational capabilities before entering the FC system). A participant is asked to produce a string prefixed with several zeros with a given input in the traditional PoW process. In PoW, the participant must perform a variety of calculations in order to produce such a series (e.g., hash operation). If the participant can generate the string in a predefined amount of time, it is a sufficiently powerful prover. However, since the participant may assign the job to other nodes, the standard PoW mechanism is incapable of distinguishing if the string is calculated by the prover itself. This limitation can be addressed by the DPPoW mechanism [59].

Chen et al. [59] developed a DPPoW-based resource authentication mechanism. As the likelihood of winning is mostly controlled by each node's computing power, only one of the nodes in the system should be picked to check the processing power of the fog device that desires to enter the system in order to prevent a collision attack. To choose which node is the winner, PoS was used [59]. To begin the DPPoW

process, one of the fog nodes is chosen to create a randomized message and request that the other nodes create a string, which is a signature with n-bit zeros over a predetermined amount of time [59]. The more computationally powerful a participant is; the quicker proof can be generated for a given complexity.

5.1.3. Enhanced PoW (ePoW)

To guarantee trustworthiness across IoT nodes at the device layer of the IoT architecture, Kumar et al. [96] introduced a trustworthy privacy-preserving approach. EPoW was developed to conduct data authentication and guard against data poisoning attacks on the original data [96]. Since the traditional PoW requires finding high proofs and hash consistency in the network requires varying levels of difficulty, the ePoW mechanism was proposed. When it comes to creating proofs and preserving the integrity of the hash chain, ePoW requires less processing power. Three functions are embedded in ePow: block generation, PoW improvement, and adding new blocks to the ledger [96]. Kumar et al. [82] suggested a system for data exchange in industrial healthcare combining BC and deep learning techniques. The network's data transactions were verified using an ePoW consensus that is based on smart contracts to create proofs and preserve the integrity of the hash chain [96].

5.1.4. Groupchain

Lei et al. [50] proposed the Groupchain (i.e., an FC-compatible expandable public BC with a two-chain structure). Groupchain is based on Bitcoin-NG, which extends PoW to enable the consensus on representative selection and transaction sequencing [50]. The goal of Bitcoin-NG is to address the issue of PoW's high latency and low throughput. Bitcoin-NG uses the PoW mechanism to choose the representative; however, while transactions are taken into account while calculating a block's hash value in Bitcoin, carrying transactions are not taken into account during the mining process in Bitcoin-NG [50]. The disadvantage of Bitcoin's inconsistent behavior is still present in Bitcoin-NG. As a result, Groupchain performs the consensus protocol in the leader's group, which significantly lowers the consensus delay, rather than gaining consensus for block data across the whole network in Bitcoin.

In order to increase pivotal impact, Groupchain utilizes the representative group to submit transactions jointly. To commit transactions, the n rolling consensus nodes comprised of leader members work together to add new blocks to the BC. Only one node from this set of the leader's group acts as the leader, suggesting candidate blocks made up of several sequential transactions. In every consensus cycle, a leader-driven candidate block is decided by a vote of the whole leader group [50]. To sum up, Groupchain offers two methods for choosing a leader in various circumstances; PoW for leader competitions among groups other than the present leader group, and round-robin among members of the stable leader group. Regardless of whether the new leader is chosen by producing blocks or by taking turns, Groupchain requires a necessary view modification [50].

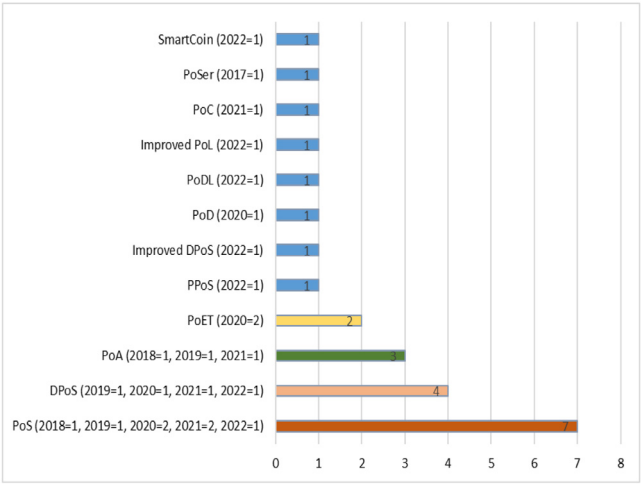

Fig. 6. Capability-proof-based consensus mechanisms number.

5.2. Capability-based mechanisms

Numerous consensus mechanisms have been presented in the literature to choose a participant based on capability due to the high energy usage of work-proof-based mechanisms. A participant's capability can be determined using a variety of variables, including their contributions, the number of cryptocurrencies they hold, the storage capacity they own, and the network's trust in them [40]. The PoS is one of the most well-known capability mechanisms. In a PoS mechanism, nodes fight for mining or leading privileges using the tokens they individually own. Because of this, the fundamental tenet of PoS is that a node with a bigger stake has a higher chance of winning the mining right. To compete for mining rights, all capability mechanisms depend on the stake. Table 4 summarizes the capability-based consensus mechanisms identified in this SLR; their characteristics, the purposes of the consensus mechanisms, and the application of the article.

Numerous consensus mechanisms have been influenced by the initial PoS mechanism. We outline the various capability-based consensus mechanisms utilized in the BC-based FC systems in this section. Fig. 6 illustrates the distribution and prevalence of PoS and its variant mechanisms. Despite facing criticism, the figure shows PoS's dominance as a consensus mechanism from 2018 to 2022, with a total of seven studies reporting its adoption in planned solutions. Following PoS, three studies utilized PoA, and four investigations employed DPoS. In 2021 and 2022, several new consensus mechanisms, including PoL, PoDL, and SmartCoin, were proposed, as indicated in the distribution shown in Fig. 6.

5.2.1. Proof-of-stake (PoS)

In 2011, the PoS mechanism was developed as an alternative to the PoW to overcome the energy consumption issue [18]. Coin age, a PoS node property, is a function of the number of coins a node holds and the length of time it has been holding those coins. The ability to load a block is no longer only determined by CPU power; rather, the higher

Table 4
Capability-based consensus mechanisms.

Mechanism	PoW characteristics	Study	Purpose of consensus mechanism	Application
PoS	• Low energy and computation power compared to PoW • High speed and throughput • Incentive fairness between nodes	[61,62,80,94]	Consensus for secure data sharing	• FC resource sharing [62] • Electric vehicle data privacy [61] • Medical IoT health sensors [94] • Edge computing market [80]
		[76]	Hybrid of PoW and PoS to enhance trust for IoT applications	• Vehicle network trust management
DPoS	• Pre-selected nodes or trustworthy observers • Low computational requirements compared to PoS	[63]	Consensus for data exchange	• Medical data exchange
		[107]	Resources contribution consensus in FC nodes	• FC resource management
		[102]	Consensus for new block validation	• Mobile edge computing resource management
PPoS	• High throughput and scalability	[79]	Consensus for new block validation	• Industrial IoT attack detection
Improved DPoS	• Delegate votes more quickly • Fix the industrial IoT's centralized security issues • Resolve the industrial IoT's data transfer and energy consumption issues	[114]	Consensus for new block validating	• Industrial IoT data security
PoET	• Works in both permissioned and public BC • Decrease participation cost • Requires far less energy than PoW	[116]	Consensus for secure data sharing	• Vehicle FC data sharing
		[71]	Consensus for new block validation	• FC trust management
PoA	• Distributed validators • Low energy consumption • Encourage rewarding BC stakeholders instead of punishing passive stakeholders • Good choice when no trust exists between nodes	[92]	Consensus for new block validation	• FC/IoT data sharing
		[68]	Consensus for resource orchestration	• FC/IoT data sharing
		[112]	Consensus for services' provisioning	• FC/IoT data sharing
PoSer	• Built based on PoW and PoS	[69]	Ensuring associated token exchanges can occur between IoT devices in the BC	• Cloud/FC/IoT data security
PoD	• Built based on PoS, but it depends on the online duration time as a stake • It enhances transaction verification performance and minimizes resource usage when compared to PoW, but similar to PoS	[60]	Consensus for new record storage	• Electric vehicle data privacy
Optimized PoDL	• Use the available computational power of the miners • Save power and communication compared to PoW	[122]	Participant nodes to train the fusion model rather than producing pointless hash values	• Medical image fusion privacy

(*continued on next page*)

a node's coin age, the more likely it is that it will be granted the privilege to load a block [127]. The main notion is that nodes with a higher stake will be the leader to add blocks to the network more frequently. The amount of energy required to mine a block will be greatly lowered as a result of this. A node's coin age must be reset once it successfully packs a block and must be reaccumulated. The security of PoS is based on the fact that nodes with substantial assets are reluctant to thwart because malicious attempts will not bring in enough money to make up for their financial loss [127]. The two best-known PoS-based platforms are Ouroboros and Casper [15]. By using a safe coin-flipping mechanism and effective time

Table 4 (*continued*).

PoL	• Uphold the stability of BC and stop wasting the processing power	[111]	Enhancing and optimizing the resources' utilization of edge computing	• Mobile edge computing resource allocation
PoC	• Cheap, efficient, distributed • Effective in detecting attacks like DoS	[48]	Identifying DoS and other attacks	• Vehicle cloud/FC data sharing
SmartCoin	• Compared to PoW and PBFT, the time required for block formation and verification is not very long	[103]	Consensus for choosing the next block creator	• Edge/IoT data sharing

slot synchronization, Ouroboros chooses the stakeholder at random. Casper gives a shakier assurance as to how much stake the attacker has in order to create interruption [127].

To achieve consensus for data sharing, Gao et al. [62] deployed PoS in the Ethereum platform with smart contracts. The choice of PoS was because it has a low computational resource consumption and high throughput compared to PoW, it maintains the incentive structure and assures node fairness, and it best meets the criteria of the proposed architecture [62]. To save carpooling records, Li et al. [61] presented a privacy-protecting carpooling solution. PoS was deployed as it randomly chooses one of the nodes to share the data. Similarly, data streaming from medical IoT sensors is processed using a light-adapted PoS consensus technique in FC [94]. Moreover, a trustworthiness model was developed by Du et al. [80] for the edge computing market to assess network entities' performance during transactions. Transactions' verification and distribution of the block creation incentive were made possible by the adoption of the PoS mechanism [80]. On the other hand, to improve trust management in-vehicle networks, Yang et al. [76] employed a modified PoS (a hybrid of classic PoS and PoW). The modified PoS is employed for IoT applications in this case, where the stakes are represented by the total amount of trust value [76].

5.2.2. Delegated proof-of-stake (DPoS)

In PoW-based and PoS-based mechanisms, BC is ruled by the richest members, which pressed for the creation of a more federated new mechanism [128]. To avoid this issue and provide a fair incentive process, the DPoS was proposed. DPoS mechanism also lessens the monopoly risk associated with PoS. In DPoS, the stakeholder allows other nodes to cast votes for verified participants, with the mining privileges alternately going to the highest participants. Accordingly, the DPoS mechanism can significantly improve authentication and consume less energy as well as provide a solution for security challenges [114]. Many experts refer to it as a representative variant of the PoS consensus system because it is based on a voting process and elect's delegates rather than individual network nodes [67]. Experts assume DPoS can manage a higher transaction volume and quicker confirmation times than PoW and PoS consensus systems since only a limited number of network nodes or trustworthy observers are needed to validate data in the creation of the new block [63].

To facilitate the exchange of medical data across various entities, DPoS was utilized in [63]. The representatives who handle the permits for authentication and accounting in DPoS are chosen by stakeholders [63]. In order to contribute resources to FC nodes, Wang et al. [107] presented a framework based on BC technology. The DPoS was employed due to its slight computing needs, compared to PoS [107]. To eliminate bid-rigging tactics in the distribution of resources for mobile BC edge computing, Qiu and Li [102] devised an auction mechanism depending on DPoS to validate the new block [102]. He et al. [74] presented a scheme for the allocation of resources in IoT using a deep reinforcement learning technique for BC-based edge computing. The DPoS consensus mechanism was applied within a private BC network to choose the node that would execute the transaction. It was recommended to use PoW or PoS to confirm transactions [74].

5.2.3. Algorand pure proof-of-stake (PPoS)

Abdel-Basset et al. [79] suggested a solution for detecting privacy breaches in industrial IoT by deploying a federated learning technique. As a consensus mechanism, Algorand PPoS was utilized to ensure that the transactions were verified [79]. The PPoS mechanism, which is characterized by high throughput and scalability properties, was suggested to boost the consensus throughput issue in PoS [79]. The chance that a participant will get selected depends on how much of their stake is spread across all tokens. Moreover, the committee members use a gossip protocol to circulate the created block among the neighbors to reach a consensus [79]. In Algorand's PPoS mechanism, the integrity of the overwhelming majority of participants is tied to the system's overall security. The system is secure when most of the stakes are in trustworthy hands. Moreover, it is difficult for the holders of a small number of stakes to negatively affect the entire system, and it would be unwise for the holders of the majority of stakes to act foolishly because doing so would weaken their position and eventually reduce the value of their holdings [39].

5.2.4. Improved DPoS

An enhanced industrial IoT network using the DPoS mechanism that syndicates BC and AI for real-time data transfer was reported by Sasikumar et al. [114]. Despite the advantages offered by DPoS, establishing fairness by relying entirely on DPoS is difficult because it will only permit those with more wealth to cast votes [114]. Also, since voting options and vote calculations are inaccurate, it is unbearable to choose the best delegates for block development in DPoS. Accordingly, to pick delegate votes more quickly and keep block data in the

trade node, an improved DPoS was proposed. The improved DPoS mechanism consists of three stages. The first stage is the creation of transactions, which is the responsibility of trading nodes, whereas the creation and verification of blocks is the responsibility of consensus nodes. A collection of delegate phrases is produced for each voting node by the first voting system, known as the honorable voting system [114]. The second stage entails developing a better voting function that will be applied to decide the value of each node. The more honor votes a node receives, the higher its chances to be elected as a delegate. The last stage in the voting system is to ascertain the deviation degree [114].

5.2.5. Proof of elapsed time (PoET)

Many private BCs, such as Hyperledger Sawtooth, are now using PoET instead of PoW because it relies on a randomized timer mechanism for network participants rather than mining hardware designed to lower complexity and increase responsiveness [129]. Each BC node in the network must wait a predetermined period, and whoever has the most completed time wins and validates the new block [71]. The node that gets alert first is the consensus round's representative and earns a reward in the form of a priority that may be used in subsequent rounds of the dispersed scheduling process. This qualifies PoET for opportunistic kinds of networks where clients are resource restricted [116].

In vehicular FC, Bonadio et al. [116] employed the PoET mechanism to bring vehicles that were exchanging information into consensus. In terms of scalability and latency, the consensus evaluation revealed that PoET performed better than PoW. Iqbal et al. [71] proposed a safe FC where tasks are transferred from roadside units to adjacent fog vehicles based on reputation ratings kept at BC. As a result, the decision model may choose a reliable vehicle for any forthcoming transactions. The consortium's members agree on the arrangement of the blocks in order to maintain uniformity. The reputational ratings of the fog vehicles were validated using PoET to ensure efficient work offloading.

5.2.6. Proof of authority (PoA)

PoA was created by Ethereum and is regarded as an effective mechanism for private BCs since it is built based on the valuation of the participant's identity and reputation in a network. As a consequence, PoA-based BC is secured by representative trustworthy nodes that are randomly selected [112]. PoA has been suggested as a potential solution to address the PoS issue of accumulating nodes that may be not even part of the network due to its stakes [130]. PoA encourages BC ownership and service by rewarding active stakeholders rather than punishing inactive stakeholders. In settings where there is no trust between network nodes, especially for private BCs, the PoA mechanism is a viable choice for reaching a consensus [68]. In PoA-based BC, representatives need to maintain and secure their computers to ensure they are not compromised. Any node has the right to be the representative providing the identity reputation. Here, participants will not compromise their identity with a negative reputation, which makes PoA more robust than PoS, for example. Moreover, in the case of attacks, only validator nodes may be compromised since PoA does not require consecutive block consensus from all validators.

Núñez-Gómez et al. [68] utilized PoA to develop the Heterogeneous, Interoperable, and DistRibuted Architecture (HIDRA), which is a decentralized FC/IoT architecture that combines permissioned BC networks with lightweight container-based virtualization solutions. This architecture aims to manage resource orchestration. The PoA mechanism was used to achieve consensus among participant nodes to enhance fault tolerance and security levels [68]. Zeigler et al. [92] presented a scheme that unifies BC and FC using the Plasma framework. The Plasma framework has the benefit of offering scalability based on off-chain or side-chain methods. Since the proposed design only requires one validator, PoA was utilized as a consensus mechanism [92]. Moreover, a provisioning approach was developed by Xu et al. [112] to guard against unsafe external service codes from dubious edge servers. Using edge servers' authentication and service validation, BC was utilized to maintain all of the off-chain services' legitimate statuses. PoA consensus mechanism adoption was also made to guarantee low latency and high throughput [112].

5.2.7. Proof-of-service (PoSer)

In their article, Sharma et al. [69] suggested a PoSer consensus mechanism that complies with the requirements of their SDN-based proposed solution to the security and privacy-related FC problems. These requirements include the nodes' participation in an activity that takes place outside the BC, such as the transmission or supply of data or the execution of a computation that will result in the exchange of tokens among participants. In order to establish the accuracy of the contribution and the viability of connected token trades in the BC, PoSer was deployed instead of the more often used PoW or PoS consensus mechanisms [69]. The 2-hop BC approach, which integrates the PoW and PoS mechanisms, was utilized by PoSer. If the genuine participant nodes have more control over the pooled resources, which include processing power and stakes, the PoSer-based BC will be more secure [69].

5.2.8. Proof-of-online duration (PoD)

Li et al. [60] proposed the PoD, as the fundamental consensus method in the proposed privacy-preserving charging strategy for electric cars, using Hyperledger-Fabric as the execution platform. When compared to PoW, PoD can significantly decrease resource usage and increase the efficiency of transaction verification [60]. The PoD is based on PoS but depends on the online duration that operating systems supply without a time-consuming stake proportion calculating method. That is, the device that can stay online for a longer period is in better functioning condition and more reliable to create blocks [60]. The analysis of PoD's resource usage in comparison to PoW and PoS revealed that PoD uses about the same number of resources as PoS and significantly less than PoW. However, compared to the PoS, the PoD uses the amount of time that devices are online as stakes, which lowers

the computation of virtual resources [60]. Moreover, PoD has fewer decentralization properties than PoS, but its consensus effectiveness is a little bit greater [60].

5.2.9. Optimized proof-of-deep learning (PoDL)

An inception network and convolutional neural network-based fusion of medical image model was developed by Xiang et al. [122]. Xiang et al. [122] developed an improved PoDL mechanism that requires participant nodes to train the fusion model rather than producing pointless hash values as is done in PoW-based BC to save computational power. PoDL solves the issue of power consumption by using the computational capacity of nodes to do practical deep-learning tasks as proof rather than searching for a nonce as in PoW. The original PoDL does not support multitasking and requires blockheads to be submitted twice, which increases communication costs. Accordingly, the improved PoDL extends PoDL for multi-task scenarios and optimizes PoDL's workload.

The PoDL process, in Xiang et al. [122] design, was done over two steps. The task presenter (i.e., institutions with limited computational power) distributes medical fusion tasks (containing task information, fusion models, training dataset, and so forth) to full nodes (i.e., a strong network of trustworthy institutions) and miners in the first phase (i.e., institutions with strong computational resources). After selecting the job with the greatest return, the miners train the fusion model as necessary and provide the hash values, by the end of this step. The task publisher sends a link of the dataset to the full nodes and other miners in the second stage, who then test their trained models and create blocks before submitting their test findings to the full nodes by the end of stage 2. The successful miner wins the posting of the new block once the block with the highest testing results is accepted by full nodes. Full nodes disregard models whose step 1 hash is not obtained [122].

5.2.10. Improved proof-of-learning (PoL)

Zheng et al. [111] presented a model to enhance resource allocation in BC-based mobile edge computing. The model was implemented with PoL consensus, which does not squander the CPU power of edge computing servers. The challenge of neural network training takes the role of the hash puzzle in PoL. Every node also serves as a participant in the BC system, which means that nodes produce blocks after completing the training neural network task that has been issued by the PoL consensus process. Each support node locally trains the model in a given amount of time by simply substituting the hash puzzles with the neural network training task. The support node must include the learning outcomes in the transaction when a training assignment is complete. By using the loss functions and output sets of the neural network, additional nodes can use this method to determine whether the training result is accurate. PoL can prevent misuse attacks in this way while also conserving computational resources [111]. However, because PoL sets a training time, the speed of block production cannot change in response to BC network capacity. To solve this problem, an improved PoL was proposed by [111] such that the node just has to finish the specified number of training cycles based on the specified set of data, where the specified number denotes the level of complexity of the training. To alter the speed of block production, the system can vary the complexity level [111].

5.2.11. Proof-of-creditability (PoC)

As a foundational consensus mechanism for the identification and banning of false information, each node in the network is seen by PoC as a peer that adds to a distributed ledger [131]. After conforming to a pre-defined number of mistakes that should be contained in each block, it is added to the BC network. Finally, all peers are informed of the detection process carried out by PoC. PoC allows mining nodes to validate the transactions using their hardware [131]. PoC maintains a list of potential solutions in the mining nodes, which is opposite to PoW where participants use the mining device's process computation or PoS where participants follow the participant's stake. So, the more power provided by the node, the greater number of possible solutions stored on it. This increases the likelihood of capturing the needed hash value from its list, and then higher chances to win the mining incentive. Accordingly, PoC is more efficient than PoW and PoS. Lakhan et al. [48] offered a scheme for task scheduling and offloading in vehicular FC. In this scheme, the authors used both PoW and PoC consensus mechanisms. To complete data offloading while in mobility, nodes use PoW to interact with one another. To identify DoS and other attack-related problems, the PoC was used [48].

5.2.12. SmartCoin mechanism

An incentive scheme for vehicles based on consortium BC was proposed by Vishwakarma and Das [103], which they called "SmartCoin". In the consortium BC-based SmartCoin, the roadside unit of the VANET network serves as the public network while the monitoring node was made into a private BC. This helps to confirm the collected data. According to the suggested approach, the message source vehicle receives a rating from the vehicles depending on the message's veracity [103]. The vehicle network benefits from decentralization, transparency, and immutability provided by BC. The round-robin process is used by SmartCoin to quickly choose the next block creator.

Blocks are generated by SmartCoin at regular periods. When the period is finished, the following node in the network begins producing blocks. Since there is no possibility of two participant nodes producing the same block at the same time, the issue of unnecessary fork generation has been resolved with SmartCoin [103]. The network will not be impacted even if one node is not operating properly or fails to operate at all. Thus, failing of any node does not influence the operation of the SmartCoin, and hence it is fault-tolerant. The block is disseminated around the network for validation by the leader mining node. The blocks are verified by each node in the network, which then marks them as trustworthy. If there are N nodes, the leader node does not add the blocks until it has 3N/4 valid responses. SmartCoin performance is improved as 90% less consensus latency and 70%–80% lower storage and transmission costs than PoW and PBFT-based BC systems [103].

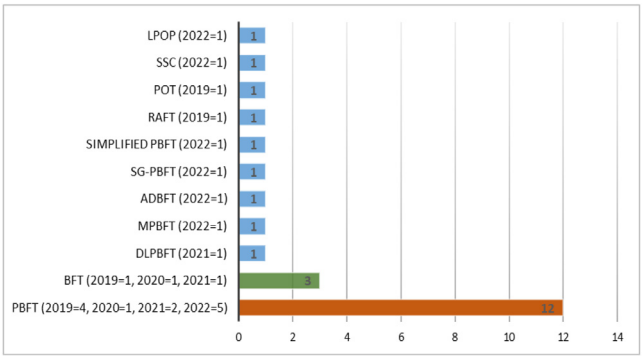

Fig. 7. Voting-based consensus mechanisms number.

5.3. Voting-based mechanisms

Using a voting procedure, the voting-based mechanisms select a participant responsible for creating a block. Voting-based mechanisms were developed to solve the problem of PoW-based mechanisms' high energy usage. As the choice is made according to financial domination, the voting mechanisms also deal with regard to the affluent growing richer in capability-based mechanisms. By considering that there would be distinct node breakdowns in the network or that certain nodes may act fraudulently, the voting mechanisms are built to accommodate Byzantine faults. Further divisions of voting-based techniques include crash fault tolerance-based and BFT-based. The consensus that is based on BFT eliminates the possibility of malicious and failing nodes, such as BFT, PBFT, mPBFT, and DLPBFT mechanisms. Contrarily, crash fault tolerance-based consensus just guards against the possibility of failing or crashing nodes. Crash fault tolerance consensus is demonstrated through the RAFT, Paxos, and Federated mechanisms [40]. However, crash fault tolerance mechanisms are unable to resolve the Byzantine fault-tolerance issue, which arises when nodes exhibit malevolent activities. Other consensus voting mechanisms such as DBFT and Algorand are built to elect the committee members, from which the representative or leader is elected, in addition to block-adding phases. Typically, the representative node and committee nodes are chosen during the first phase (only committee nodes are eligible to participate in the consensus-building process). Table 5 summarizes the voting-based consensus mechanisms identified in this SLR; their characteristics, the purposes of the consensus mechanisms, and the application of the study.

We outline the various voting-based consensus mechanisms utilized in the BC-based FC systems in this section. Fig. 7 portrays the distribution and prevalence of BFT and its variant mechanisms. Notwithstanding the criticism of high communication costs, PBFT stands out with the highest frequency in this SLR, being utilized in 12 studies, and among them, 5 studies were published in 2022. Additionally, most of the studies published in 2022 proposed PBFT or a modification of PBFT, such as LPoP, TPM, and SG-PBFT.

5.3.1. Byzantine fault-tolerant (BFT)

The BFT is one of the Hyperledger mechanisms, where users are allowed to use their consensus structures [66]. BFT is designed for distributed networks to achieve consensus even if some of the participating nodes fail to validate the transaction [91]. In BFT, all participant nodes must concur on a unified state to prevent total failure while presuming that some of the nodes may be untrustworthy. However, only one validator manages the transaction—selected by a majority of the participants [132]. Since BFT demands that all network nodes take part in consensus, it does not apply to public BC that have a significant number of dynamically connecting nodes [102]. It works well for consortium BC and private BC with a limited number of permanent nodes and minimal delay [102].

A consensus architecture built on the Byzantine method was suggested by Sheikh et al. [77] to improve the data sharing security among electric vehicles as well as to protect the system from fraudulent intrusions [77]. Moreover, BFT was utilized in [65] to achieve the consensus on a device authentication request. Following receipt of the requested findings from complete nodes, the participant node compiles them to determine the device's public key and identification cancelation status before returning the device's authentication outcome [65]. A privacy-preserving method using BFT consensus was proposed by [64] to improve verified data sharing in vehicle FC. Under this method, a selected few participants create the necessary blocks and disseminate them to the whole network. A particular block is regarded as validated and entered into the chain if the number of confirmations meets a certain threshold [64].

5.3.2. Practical BFT (PBFT)

The PBFT is a BFT extension that can function efficiently in asynchronous systems and is described with low latency [64]. PBFT was created to address some of the BFT's shortcomings. Compared to BFT, PBFT is a protocol with lower computational complexity and a higher degree of usefulness in distributed environments [91]. PBFT is implemented on some BC platforms, like Hyperledger. The maximum number of malicious nodes cannot be more than a third of all nodes, according to PBFT [90]. The pre-preparation stage, prepare stage and commit stage are the three key stages of PBFT [127].

The representative node transmits a block to regular participating nodes in the initial pre-prepare stage so they can validate it. Each regular node transmits the confirmation results to all other nodes during the second preparation stage. In the third stage, each node sends the preparation stage verification results to all other nodes once more. Based on the messages received, each node then performs a final block confirmation. Adding the block to the BC is possible if the "YES" votes number is more than 2f+1. Each block is validated in real-time by the majority of nodes during the block-adding phase of PBFT, which indicates that in the PBFT mechanism, every two nodes must interact with one another [95]. As a result, while PBFT significantly improves crash resistance to Byzantine robustness, it also necessitates multiple communications among each pair of nodes, compromising communicational

Table 5
Voting-based consensus mechanisms.

Mechanism	Characteristics	Study	Purpose of consensus mechanism	Application
BFT	• Low variance of reward • Less latency and energy usage than PoW	[65,77]	Consensus for new block validation	• Electric vehicle energy trading [77] • FC/IoT authentication [65]
		[64]	Consensus for data sharing	• Vehicle FC data sharing [64]
PBFT	• High communication costs • Low variance of the reward and energy consumption • It solves the BFT generals' problem for asynchronous environments • It maintains stability even if the number of malicious validators rises	[66,85,90, 91,95]	Consensus for authentication	• Vehicle FC authentication [66,90,95] • FC rogue nodes protection [91] • Cloud/FC/IoT data sharing [85]
		[75,78,83, 101,108, 115]	Consensus for new block validation	• Multi-access edge computing task sharing [75] • Industrial IoT resource allocation [78] • 6G-VANET data sharing [115] • Fertility preservation [108] • Cloud/FC/IoT data sharing [83] • Multi-access edge computing-VANET task sharing [101]
		[45]	Consensus for confirming a block's accuracy	• Vehicle FC-6G data sharing
DLPBFT	• Low transaction processing time compared to PBFT • It ensures the accuracy of the data recorded on the BC • Higher throughput compared to PBFT	[67]	Consensus for data sharing authentication	• Edge computing-as-a-service scheduling data sharing in industrial energy
mPBFT	• It is appropriate for a network of vehicles with SDN capability that utilizes permissioned BC	[87]	Consensus for choosing the leader	• Vehicle network with SDN support data storage
AdBFT	• Calculate the task participants' current reputation	[105]	Verifying miners' credentials and screening the data	• Edge mobile crowdsourcing data storage
SG-PBFT	• Improve the security and efficiency PBFT	[41]	Consensus for new block validation	• Vehicle FC data sharing
Simplified PBFT	• Simplify complex operations PBFT consensus mechanism	[106]	Consensus for new block validation	• Industrial IoT data sharing
RAFT	• Scalable, lightweight, and high throughput • It provides a stable response time • It meets the requirements of a small number of firms	[72]	Consensus for data sharing authentication	• 5G mobile edge computing authentication

(*continued on next page*)

sophistication while saving computational costs, which slows consensus down as the number of nodes rises [127].

Yao et al. [66] created a simple anonymized authentication scheme for vehicular FC. The PBFT mechanism was used in this scheme to get a consensus on the new transaction. PBFT was chosen over other mechanisms like PoW, PoS, and DPoS because of its medium energy usage and a relatively smaller number of malicious nodes (i.e., 33% in PBFT compared to 51% in PoW and PoS) [66]. In a similar vein, Kaur et al. [90] presented a lightweight authentication system for vehicular FC. With the use of PBFT, participants validate transactions [90]. Alshehri and Panda [91] and Eddine et al. [95] provided a method to prevent malicious fog node usage, and the PBFT mechanism was utilized to authenticate

Table 5 (*continued*).

PoT	• High level of trust among pre-selected nodes • Light compared to PoW, PoA, and PoS • It eliminates malicious nodes and effectively reduces consensus time	[121]	Consensus for new block validation	• Edge computing data sharing
SSC	• It utilizes hardware-based security to prevent identity theft	[117]	Selecting miners in a stochastic way to ensure dispersed but timely consensus	• Edge/drone data sharing
LPoP	• It creates a group of verifications • It enables many sensors to exchange data in a lightweight way	[97]	Lightweight consensus to carry out heavy mining	• Industrial IoT auction fair bidding process

data shared. Okegbile et al. [85] developed a method for securing data exchange in cloud/edge-IoT networks utilizing a collaborative approach, where many data sources and data consumers work together to complete data exchange. PBFT was utilized to validate the data shared and create trustworthy validation procedures while preventing the single-node failure threat [85].

Rivera et al. [75] provided a framework to offer a trustworthy cooperation mechanism amongst edge servers. Each new transaction is disseminated to all nodes, and before each node performs the transaction and creates a block, the consensus is reached using the PBFT mechanism [75]. A method for exchanging data was proposed by Wang et al. [115]. The private set intersection protocol and smart contracts were used to create the PBFT mechanism. Voters are chosen using the private set intersection protocol mechanism, and the voting procedure is carried out using smart contracts. PBFT is used to allow selected nodes to carry out the consensus process instead of all nodes, which considerably enhances the network's overall throughput [115]. Li et al. [83] suggested an approach for enhancing collaborative resource allocation to IoT devices utilizing the reinforcement learning method for cloud/edge-IoT networks. The transactions that are added to the block are first verified using the PBFT consensus mechanism [83]. Similar to this, Yang et al. [78] suggested a deep reinforcement learning-based model for resource allocation and energy conservation in industrial IoT. To reach a consensus on the released transaction, the PBFT mechanism was used [78]. Poongodi et al. [101] developed a neuro-fuzzy systems-based BC strategy for VANET networks to improve data security. The identification and key exchange method suggested in this work were employed by an authorized IoT device to add blocks without coming into touch with vehicle users. PBFT was used to ensure the transaction's validation [101].

In order to increase customer engagement and satisfaction, Liao [108] presented a federated BC-based egg banking scheme for fertility preservation. The addition of a new block made use of the PBFT consensus mechanism. Only block representatives are permitted to take part in the consensus procedure. This concept offers a reward to the block's primary representative for each new block that is created [108]. Gao et al. [45] developed a model to improve security and trust in VANET by merging BC and SDN in FC. To confirm a block's accuracy, the PBFT consensus mechanism was used. After receiving the block, a few pre-selected nodes take part in voting until a consensus is formed to choose a representative, which then generates a block [45].

5.3.3. Double-level PBFT (DLPBFT)

Due to problems with the PBFT consensus mechanism's actual implementation, other academics have proposed advanced forms of PBFT. The PBFT consensus mechanism's very high communication cost makes it unsuitable for dynamic systems [41]. The DLPBFT is a new extension of PBFT, which has less communication overhead than PBFT, and accordingly, the consensus reliability is more for DLPBFT [67]. The PBFT mechanism takes longer to validate transactions than the DLPBFT mechanism. DLPBFT reduces the storage pressure of a single BC and increases computational performance.

Bai et al. [67] used the DLPBFT mechanism for EdgeChain to ensure data consistency and decrease the processing time for data stored in BCs. The DLPBFT consensus method was used to vote for the calculation messages, which include data and outcomes from Stackelberg game optimization, and then offer traceability and protect data secrecy. The top 80% of nodes in the first level act as regular participating nodes and cast votes on messages received through leader peer streaming; the remaining nodes act as storage nodes. The regular nodes at the second level are made up of each EdgeChain's major peers in order to assure node trust and increase consensus throughput by lowering the number of consensus nodes.

5.3.4. Modified PBFT (mPBFT)

Using BC, Vishwakarma et al. [87] proposed a lightweight BC-based security protocol for vehicle networks with SDN for safe storage and communication. This protocol is a permissioned-based BC that uses a new consensus mechanism, which is a modified PBFT (mPBFT) to achieve safe storage and communication for vehicle networks. The mPBFT is more appropriate for permissioned BC than other mechanisms [87]. Four states—new-round, verification, commitment, and append—are used by the mPBFT. The state's process

progresses from a new round to a committed state. In the new-round stage, a leader is chosen from the pool of miners. In the verification stage, except for the leader node, all the other miner nodes confirm the block following a vote of more than two-thirds of the mining nodes after which a commitment message is sent throughout the network. If there are more than two-thirds of the commitment messages, the block is confirmed. All of the miner nodes append the suggested block into the BC during the append stage [87]. The mPBFT consensus mechanism promotes fairness by giving each participant a similar opportunity to submit a block. In comparison to previous consensus mechanisms like PoW, PoS, and traditional PBFT, the mPBFT mechanism provides a selection of the leader, which minimizes the consensus latency and decreases computational resources and cost [87].

5.3.5. Algorand delegated PBFT (AdBFT)

Representative nodes and regular nodes are two different classes of nodes in DBFT. Based on the percentage of the stake they each own; the regular nodes participate to select the representative nodes. All nodes in the committee are representatives. The representative who creates the block is chosen among the representative nodes via polling [40]. In the representative selection stage, Algorand employs verifiable random functions to generate the representative and committee members at random for every block. Transactions may be verified once the block is uploaded to the BC, much as PBFT. The main goal of representative committee consensus mechanisms is to reduce communication overhead by limiting the number of nodes involved in the consensus process. Furthermore, since all nodes have an opportunity to join the committee, the service's decentralization is not sacrificed [39].

Wang et al. [105] suggested a data privacy protection model combining federated learning and BC for mobile crowdsourcing. All consensus nodes' credentials and data shared are verified using the AdBFT consensus mechanism, which integrates the Algorand mechanism with the DBFT mechanism [133]. The suggested solution uses the Algorand mechanism to arbitrarily choose certain nodes as interim consensus nodes and choose an interim leader node. This successfully lowers the likelihood of member inference attacks by providing the leader that analyzes data randomly each time [105]. This also can successfully address the threat of single-point failure [105]. Swarm as an off-chain storage technique was deployed to address the issue of BC's restricted block storage [105]. The evaluation of AdBFT showed that tag-flipping attacks, Byzantine attacks, and data poisoning attacks may all be effectively prevented since the system chooses task consensus nodes and task data of the highest capacity [105].

5.3.6. Score grouping-PBFT (SG-PBFT)

Xu et al. [41] introduced Score Grouping-PBFT (SG-PBFT), a novel consensus mechanism for distributed vehicle networks. By streamlining the PBFT consensus mechanism and employing a score grouping method to increase consensus efficacy, the SG-PBFT consensus mechanism outperforms the conventional PBFT mechanism [41]. There are N/2 consensus nodes in the SG-PBFT, which will choose the leader node. The leader shares the transaction with all consensus nodes after receiving the request from the client node. If the verification from every consensus node is successful, the message is passed to the preparation step. Consensus takes place when the leader node receives identical confirmation messages from more than 50% of the consensus nodes. The lowest-scored consensus node will be placed at the bottom of the nodes list, while the successful nodes' scores will be raised by 1 point. The results of the SG_PBFT evaluation demonstrated that the strategy may significantly increase consistency efficiency and successfully thwart single-node failure. In particular, the consensus latency of the SG-PBFT method is only around 27% of what is needed for the original PBFT mechanism when the number of consensus nodes hits 1000 [41].

5.3.7. Simplified PBFT

Yang et al. [106] proposed a distributed model for data-sharing in industrial IoT that is based on BC and edge computing to enhance scalability. The evaluation suggested that the model may significantly increase data sharing's effectiveness, dependability, and security while imposing realistic and tolerable overheads [106]. The lightweight model presented in this article does not operate with the conventional PBFT consensus mechanism due to the high communication overhead. The PBFT is therefore treated as a categorizing service, which simplifies the PBFT consensus mechanism [106]. A nonce, or incremental number, is used to verify every transaction. If the nonce offered by a node is not accepted by other nodes, no further transactions may be performed, requiring the node to concur with the other nodes [106].

The simplified PBFT mechanism's concept is fundamentally similar to that of the PBFT. Orders, endorsers, and committers make up the BC network's member nodes. The order node is in charge of packing transactions, the committer node is in charge of obligating the transaction, and the endorser node is in charge of confirming the transaction [106]. Before the committer node publishes the transaction, the endorser node validates it, and the order node then bundles the transactions following predetermined rules to produce blocks, such as the block height or timing of transactions. The packed blocks are then sorted by the order node using predetermined criteria, such as creation time. A randomized technique among a group of order nodes can be utilized when a transaction is filed to identify a specific order node to arrange the blocks. This guarantees that the transaction is consistent and prevents the bifurcation problem [106].

5.3.8. Reliable, replicated, redundant, and fault-tolerant (RAFT)

The Linux Foundation launched Hyperledger in 2016, and it is the most active and widely used permissioned BC in the industrial and IoT realms [31]. The RAFT mechanism is used in the permissioned BCs intended for enterprise environments and is a great match because it is more transparent and consumes fewer resources [31]. RAFT can withstand system failures up to N/2 by employing a "master and follower"

strategy [31]. A master node is chosen by the collection nodes (this group of nodes is known as the "consenter set"), and its decisions are replicated by the followers. Furthermore, the RAFT setup comes straight from the master node. The master node keeps track of a replicated log across all nodes. A more dispersed ordering service is made possible by this robust design [31].

One of the key objectives of RAFT is to build a distributed consensus mechanism that is more understandable than Paxos without losing efficiency or correctness. Paxos is one of the most widely used mechanisms for achieving distributed consensus for permissioned BC and many distributed consensus mechanisms are built on or inspired by Paxos [134]. Paxos, on the other hand, is a notoriously difficult mechanism to apply in order to satisfy its performance and correctness criteria. In terms of fault tolerance and performance, it is comparable to Paxos [134]. The distinction is that it can be broken down into comparatively self-contained sub-problems and covers all of the main components needed for practical systems. For supply chains operating in 5G networks, Jangirala et al. [72] presented a lightweight BC-enabled RFID-based authentication method. The needs of an internal RFID system inside a firm may be satisfied in a private BC using a RAFT consensus mechanism [72]. Using RAFT, it is possible to safeguard the secret tag data while sharing non-sensitive tag data for authentication [72].

5.3.9. Proof-of-trust (PoT)

Jayasinghe et al. [121] proposed the TrustChain service-a privacy preserving BC-based scheme, in which PoT consensus is used. PoT is a substitute for traditional consensus mechanisms such as PoW and PoS when a node's record on a network increase in value with its participation in the BC network. When a node attempts to tamper with the system, trust is undermined. Maintaining a reasonable conversion factor between trustworthiness and the network's unit of value (such as a token or coin) is crucial in PoT networks so that trust may be properly taken into account during the node selection procedure for block verification [121].

PoT mechanism chooses a set of nodes as consensus participants based on their ability to sustain greater levels of trustworthiness [121]. The PoT uses a voting mechanism reliant on reputation, which is driven by the BFT-based mechanism. The selection of participants in PoT, however, is not governed by a single participant and enables any node with a sufficient level of trustworthiness to be chosen as a leader [121]. There is no centralized master authority, thus anybody may start spinning and take part in consensus, which increases decentralization. A malevolent participant would find it challenging to influence the voting process since many new nodes may be connected to the system [121]. Through the TrustChain platform, nodes may gain the trust of other nodes by participating in transactions that provide value to the ecosystem. The evaluation revealed that TrustChain can conserve resources in IoT networks and reduce latency and security problems related to centralized consensus mechanisms such as Pow and PoS.

5.3.10. Stochastic selective consensus (SSC)

By separating the data component (also known as the block ledger) from the block header and moving it to off-chain storage, Singh et al. [117] developed BC-based architecture for the drone network. The use of a trusted platform module in the approach adds chip-level protection to the security keys kept on drones so that if a drone is stolen, an attacker will not try to alter it without being aware of its trackability. Additionally, the BC only consists of block headers, which improves the efficiency of communication, synchronizing, and processing. To guarantee that each drone is in charge of its block, Singh et al. [117] developed a lightweight consensus technique (SSC) utilizing stochastic sampling and transaction signatures. According to the findings, selecting a voter takes less time when there are fewer voters. However, the time increases with more drones [117].

5.3.11. Lightweight proof-of-proximity (LPoP)

Bhattacharya et al. [97] presented a BC-based mining-as-a-service model for industrial IoT and proposed an auction technique for fair bidding amongst participant nodes. A novel consensus technique—LPoP was proposed to create group verifications rather than single block verification for data sharing and enable many sensors to communicate the data in a lightweight way [97]. The foundation of LPoP is the Delegated PoP (DPoP) consensus mechanism [135]. The fundamental principle of DPoP is that nodes vote for their delegate utilizing their stake, which is based on their proximity to a sensory event, rather than on the validity of a specific block [135]. Delegates are in charge of verifying transactions and selecting the block sequence by contrasting their sensing event recordings to the block sequence proposed by the system. To prevent being singled out, selected delegates are randomly distributed among the nodes in the pool, ensuring that a distinct group of nodes participates in the delegated duty for each round of consensus [135]. A voting node is chosen depending on the node's proximity to the transactional occurrence in a voting-based consensus [136]. Due to the low likelihood of such nodes being chosen, this does not guarantee fairness for nodes that are too far away. Because prior methods did not take into account the remote nodes, Bhattacharya et al. [97] modified the DPoP consensus by adding a fair reward policy, which is known as LPoP [97].

6. Evaluation of the identified consensus mechanisms

A large number of processes and the wide variety of BC consensus mechanisms properties make it challenging to fully comprehend them. It may be difficult to establish whether a given mechanism fits particular criteria. In this scenario, visual assistance would be beneficial. To do this, we provide a summary that can be used to identify the optimum consensus mechanism based on certain criteria in a range of contexts. In order to achieve its goal, the summary makes use of decentralization, security, and effectiveness criteria to evaluate the consensus mechanism best match. In the sections that follow, we describe these criteria for each consensus mechanism category and then offer options for how to deploy different mechanisms in various contexts.

Table 6
Consensus mechanism performance evaluation.

Mechanism	Decentralization	Security	Effectiveness
Work-proof-based	• Decentralization risk	• Adversary 50%	• High scalability • Low throughput and high energy requirements
Capability-proof-based	• Decentralization risk	• Adversary 50%	• High scalability • Medium throughput and medium energy requirements • PoDL and PoL to save energy
Voting-based	• High decentralization	• Adversary 33%	• Low scalability • High throughput and Low energy and resource requirements • High communication cost • DLPBFT shortens the transaction time of PBFT • SG-PBFT and SSC enhance the security of PBFT • RAFT and LPoP lightweight meet IoT and small enterprise requirements

6.1. Performance analysis

Based on our review of recent surveys, the performance of BC consensus mechanisms may be assessed in three factors: decentralization, security, and effectiveness [22,39,40]. However, each of these factors has several evaluation indicators. The ability and rights of distributed nodes to participate in the consensus mechanism are referred to as decentralization. The security of the BC system relates to whether it assures the proper functioning and stability to withstand various attacks such as DoS attacks, double spending, eclipse attacks, Sybil attacks, and selfish mining [39]. Scalability, latency, throughput, and cost are among the aspects of effectiveness [40]. The number of nodes that can contribute to the consensus mechanism is referred to as scalability [50]. Latency is the amount of time it takes for a transaction to pass from the creation phase to the final confirmation phase. Throughput is the number of transactions that a system can accomplish in a given amount of time [62]. Cost relates to the cost of various resources such as power, storage, and CPU requirements [22]. The implementation of a consensus mechanism frequently necessitates a trade-off between these three factors (i.e., decentralization, security, and effectiveness). Under standard conditions, however, security should not be compromised [39]. Table 6 summarizes the performance aspects of the three mechanisms categories.

6.1.1. Decentralization

The decentralization of FC refers to the distribution and dispersal of computing resources, data processing, and decision-making capabilities across multiple decentralized nodes within an FC network that aims to bring computing resources closer to the edge of the network [20,53]. Decentralization in FC shifts computational tasks and data processing from a centralized cloud to a distributed network of edge devices, promoting efficiency, responsiveness, and scalability while addressing the unique requirements of edge computing applications [8]. When combined with BC, FC's inherent decentralization creates a number of difficulties, such as scalability problems resulting from the coordination of BC transactions across a distributed FC environment, increased latency brought on by BC consensus mechanisms, privacy concerns when integrating sensitive data with a public ledger, resource limitations on edge devices, security flaws, architectural complexity, interoperability barriers, and difficulties with regulatory compliance [137]. However, with careful planning and resolving these issues through deliberate design and technology decisions suited to particular use cases, successful integration may result in advantages like improved security and transparency [138].

Due to low throughput in work-proof-based mechanisms, for instance, a mining pool could be formed by aggregating the mining assets of multiple miners in order to maximize the likelihood of producing a new block. Once a mining pool earns a payment for generating the next block, the payment is distributed evenly among the miners. However, this has resulted in centralization issues, with block creations restricted to a small number of miners [22]. A consensus cannot be reached if validators do not act rapidly enough to ensure double-spending prevention. As a result, block construction is extremely sluggish. Furthermore, the block size is limited such that the bigger block sizes will result in more unwanted splits due to latency amongst nodes. Real-time voting is necessary during the block addition stage of voting-based mechanisms, which results in minimal transaction verification delay. Accordingly, voting-based mechanisms can achieve good decentralization.

6.1.2. Effectiveness

When used with BC, fog nodes' efficiency varies depending on the situation. On the one hand, this integration can strengthen FC systems' dependability and credibility by improving security, openness, data integrity, and decentralized consensus [139]. However, difficulties with scalability, increased latency, resource limitations, architectural complexity, and related expenses may result in a reduction in overall effectiveness, especially in latency-sensitive applications [140]. The efficacy of a particular implementation will depend on how well these issues are handled and if the advantages of improved security and transparency exceed any possible negatives [141].

Although the work-proof-based representative selection stage has a limited throughput and the possibility of centralization, it guarantees excellent scalability. On the other hand, each node in work-proof-based mechanism applications should expend energy to calculate the hash. As the network's difficulty rises, so does its energy usage. When measured throughout the

entire network of ASIC/GPU mining rigs located all over the world, the consumption of energy is rather large. The use of such a large amount of energy has sparked concerns about the long-term viability of work-proof-based mechanisms [40].

Although voting-based mechanisms can guarantee low transaction latency, it results in high communication overhead and there is a chance of DoS assaults [39]. Due to the restricted number of validating nodes, voting-based mechanisms typically face node-scalability issues. In capability-based and work-proof-based mechanism applications, there could be a significant number of participants engaged. The on-chain data-sharing system will face scalability challenges as more participants engage and information-sharing evolves. Accordingly, off-chain storage systems were proposed as an alternative to on-chain transactions in order to boost scalability [26]. Off-chain information exchange methods necessitate cross-organizational communications channels, which increases the load on the organization to create and manage these channels. Furthermore, these methods cannot ensure the privacy and integrity of an organization's data. For instance, if Organization X has to modify the original data to match the particular needs of Organization Y, then the original data may differ from the shared data with Organization Y [26].

6.1.3. Security

Another compelling reason for work-proof-based BC is its inherent security. The number of miners in work-base-proof BC is significantly bigger than the number of validators in capability-based BCs. This means that work-proof-based mechanisms are more decentralized than capability-based mechanisms. Collusion among validators in capability-based BC validators is considerably more likely than in any popular work-base-proof BC. As a result, many researchers have questioned the security of capability-based BC. The counter-argument, on the other hand, emphasized the issue of centralization, which might lead to a collusion attack in work-proof-base BC [39]. According to many authors (e.g., [79,142–146]), due to its decentralized design and the allocation of computer resources at the edge of the network, FC brings certain security needs and difficulties. Here are some significant security issues unique to FC and IoT connected to FC nodes:

- The computing capabilities of IoT devices may be constrained, and they may also be more vulnerable to physical manipulation and compromise. If IoT devices are exploited, data saved locally on those devices may be exposed. Also, having insufficient resources for complicated security mechanisms, IoT devices may be less resistant to attacks.
- Due to their deployment in the wild, IoT devices lack the physical protection afforded to servers in controlled environments. This exposure increases the risk of unauthorized individuals gaining physical access to IoT devices, potentially compromising their functionality, extracting sensitive information, or even causing intentional damage.
- Every IoT device, in an FC and BC integration, is given a distinct identity stored on the BC. An attacker using Sybil can try to get into the system by creating many false identities for harmful or nonexistent devices. This can cause phony devices to be added to the BC without authorization, which might compromise the accuracy of the device identity data. It gets more difficult to tell real devices from fakes.
- Network connectivity issues pose another significant security challenge for IoT devices, impacting their ability to maintain stable access to the global network. In a BC network, extended outages or sporadic network access might interfere with the consensus process. Consensus algorithms, like PoW and PoS, depend on constant node agreement and communication. Disrupting consensus may result in disagreements on the BC's current status, which leaves the network open to assaults like double-spending or illegal transactions. Furthermore, network connectivity problems might cause delays or errors in the authentication process, which could result in illegal access, in situations where devices need to prove their identities or validate the identities of other devices.
- Keeping track of security for a large number of IoT devices may be difficult and time-consuming.
- The variety of platforms and devices seen in FC situations makes it difficult to deploy standard security measures.
- Due to the dynamic nature of FC systems, where devices regularly enter and exit the network, it is difficult to keep security configurations up to current.
- It can be difficult to strike a balance between the demand for low latency and security since strict security measures may cause processing delays.
- A crucial design decision is whether security operations should be spread to IoT devices connected to FC or centrally managed in the cloud.
- Interoperability issues may need to be resolved in order to provide security across diverse devices and platforms.
- As there are more possible entry points for attackers, the dispersed nature of FC may expand the attack surface.

It takes a mix of strong security policies, frequent updates, monitoring, and a proactive approach to threat detection and mitigation to address these security concerns in FC systems. Designing security solutions that are compatible with the unique requirements and peculiarities of FC applications and settings is crucial. By combining distributed validation, immutability, BFT, consistency and finality, trustless verification, and cryptographic hashing features and mechanisms, consensus protocols in decentralized BC networks ensure robust data integrity, making the data stored in the blockchain reliable, tamper-resistant, and trustworthy [65]. Moreover, combining these security measures, consensus mechanisms provide a robust and resilient security foundation for BC networks. The distributed nature of consensus ensures that no single point of failure exists, making it challenging for malicious actors to compromise the BC's security [77].

Double spending: It occurs when an attacker makes a second purchase of a currency, causing a fork when the transaction is not yet finished, causing the transaction to be revoked. Double spending mostly undermines the sustainability of the BC system. As a result, double spending mostly affects BCs when transactions are not instantly validated when submitted to the BC. To prevent a double spending attack, transaction validation should be precisely constructed to avoid the risk of canceling an approved transaction [39]. Through distributed validation, transaction confirmation, consensus on the longest chain, finality, and incentives, consensus protocols prevent double-spending in decentralized BC networks, ensuring the integrity and security of digital transactions and the value stored in cryptocurrencies [87].

DoS attacks: A DoS aims at rendering a network resource or server inaccessible to its intended users by interrupting the services of a client connected to the Web, either momentarily or permanently. DoS attacks are often managed in BC systems by limiting or preventing the consequences of single-point failures on the network. Only the failure of the representative node has a substantial influence on the system. As a result, the consensus mechanism should be developed in such a way that attackers are incapable of foreseeing which node is the representative node, and a very efficient process should be designed to recover from the failure of the representative node. Choosing the representative is part of PoW and PoS mechanisms. The same is true in the case of some permissioned BC, where the representative's selection is previously known to the entire network, which makes preventing the representative from DoS difficult. In this situation, a voting process during the block addition phase to determine whether the representative is authentic or not, such as PBFT-based mechanisms, can aid in the prevention of DoS.

Sybil attacks: The Sybil attack in the BC network describes the case where a user may have multiple identities, allowing a malicious node to have multiple votes and thus gain control of the system. Sybil attack primarily undermines the equality of the BC system and raises the prospect of centralization. As a result, the Sybil attack can be avoided at the representative selection phase such that the method of picking the representative does not rely on how many identities a client has. Because the representative in work-proof-based mechanisms is chosen based on computational power, and the representative in capability-based mechanisms is tied to a user's stake, the Sybil attack can be avoided. However, because the representative is chosen through voting in voting-based mechanisms, Sybil attack is possible. Setting up an identity authentication method for consortium BCs is another technique to mitigate the Sybil attack.

Eclipse attacks: In this attack, the intruder compromises the victim's routing table to separate the victim from the authentic BC network. To run the Eclipse attack, attackers first conduct the Sybil attack in order to build up a sufficient number of Sybil nodes and proclaim them to be legitimate nodes. The Eclipse attack would not occur in BCs that use authentication mechanisms, such as consortium and private BCs; nevertheless, it is difficult to prevent an attacker from concurrently controlling several nodes in a public BC because no authentication method is used. While it is possible to create a decent system to stop Sybil attack at the consensus level, it is difficult to stop network isolation induced by Eclipse attack [39]. Eclipse attacks must be addressed at the network communication layer, such as by upgrading routing tables regularly.

Selfish mining: This attack is primarily aimed against work-proof-based mechanisms. After correctly processing one of the blocks, a selfish miner can keep mining the next ones, retaining the leadership without broadcasting the mined blocks to the remainder of the miners. The selfish miner can distribute the cleared blocks and collect rewards for them when the other miners close up on him. Timestamps may be a remedy to this occurrence. That is, if a miner publishes a large number of blocks with recorded timestamps in a single round, other miners may reject them.

6.2. Suggestions for selecting consensus mechanisms

According to Fu et al. [39], if the representative picking method is intended to be equitable and the outcome is unknown, the block adding can be streamlined, and blocks may be added without voting immediately after verification. If the block-adding method is constructed with real-time voting, blocks are verified immediately and then uploaded to the BC, hence the transaction verification step is unnecessary [39]. Work-proof-based mechanisms and capability-based mechanisms are good options if the system has to reward the participant/validating nodes. The main uses of these consensus mechanisms are public cryptocurrencies due to their underpinning incentive structures [22]. A private BC system, on the other hand, frequently does not depend on any cryptocurrencies to encourage or reward any validators to manage the BC system. In private BC networks, non-incentivized consensus methods predominate. In comparison to other kinds of consensus processes, private BCs use a relatively small number of resources and are also quite scalable. However, these approaches are more susceptible to assaults due to the relatively small number of validating nodes [22]. DPoS and PBFT variants are the recommended choices if an incentive-based mechanism is necessary for a highly scalable BC system that wants to spend less power.

DPoS and PBFT variants will, nevertheless, enjoy the previously mentioned modest security. On the other hand, PoW mechanisms are better suited if security is the top goal. There are two possibilities in this situation: memory-bound or CPU-bound. Memory-bound PoW methods should be chosen if ASIC resistance is needed. Scalability must be given up in this situation, and such systems use a lot of energy [22]. In addition, energy usage also plays a role in selecting the best consensus mechanism. While the PoS mechanism and its variants use a medium amount of energy, PoW-type mechanisms need a lot of energy. Currently, PoW-type mechanisms are extremely sluggish and can only handle a small transaction throughput.

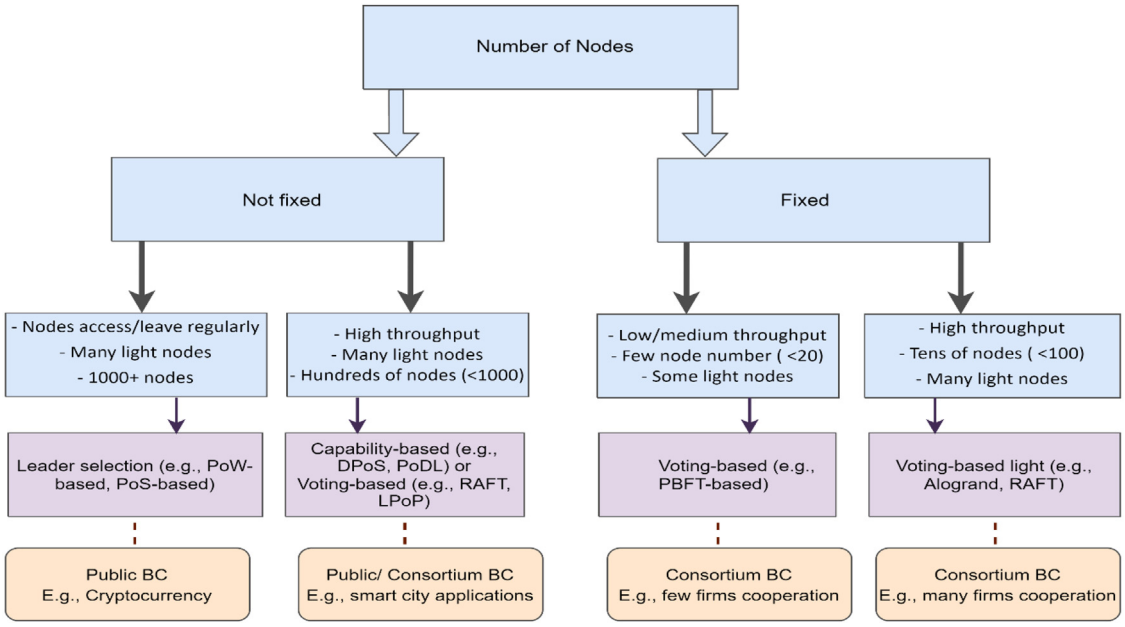

Fig. 8. BC consensus mechanisms selection decision tree.

Different consensus mechanisms are needed for various BC platforms. Work-proof-based mechanisms, for example, can fit better for public open environments BC systems, whereas capability-based mechanisms can fit better consortium BC such as business collaboration, and voting-based mechanisms can fit better private BC networks such as vehicle and drone systems. We classified the most prevalent BC application cases into four categories after assessing them. Transactional throughput, node number, and BC type were used in this classification [22,39]. Fig. 6 summarizes different scenarios and the suggested BC consensus mechanisms to best suit that scenario.

(1) When the node number is fixed, there are no weak nodes and transaction throughput is low, voting-based mechanisms may be appropriate. This instance could represent a collaboration between a limited number of organizations, with a total node number of fewer than 20, and hence significant scalability of the BC is not necessary.
(2) When the number of nodes is fixed, some nodes are light (clients manage several nodes), and transaction throughput is medium, voting-based-based mechanisms may also be appropriate. This instance may represent a collaboration between a limited number of organizations, where the total node number is also fewer than 20. Hence, the great scalability of the BC is not necessary. However, a high volume of transactions may necessitate a high transaction throughput and a short verification time.
(3) Lightweight voting consensus mechanisms such as RAFT and Alogrand may be a good choice when there are a set number of nodes—usually, more than 20 but fewer than 100, numerous light nodes, and throughput is high. However, if the number of validating nodes exceeds 20, throughput may decline due to communication overhead. It is difficult, however, to ensure the synchronization of so many full nodes in this circumstance.
(4) When the number of nodes is large and not fixed, there are many light nodes, and throughput is high, voting consensus mechanisms such as RAFT and LPoP can be a viable choice. If the needed throughput is not high, capability-based mechanisms such as optimized PoDL and DPoS may be a reasonable alternative. The massive number of sensors in smart cities is an illustration of this scenario.
(5) When the number of nodes is vast and not fixed, decentralized, many light nodes connect and leave often, and throughput is low, PoW and PoS can be a good choice. Furthermore, using procedures based on voting mode is not suggested since nodes' identities are disguised in public BC, which makes controlling these nodes very difficult. This situation can be seen when many nodes mining on Bitcoin (see Fig. 8).

7. Discussion

The purpose of this SLR is to provide answers to the research questions (RQ1: How were different BC consensus mechanisms used in FC applications? RQ2: What are the future challenges of consensus mechanisms in FC applications?). The RQ1 was addressed in Sections 5 and 6 by presenting the state-of-the-art BC consensus mechanisms used in FC and proposing a decision tree for selecting the most appropriate mechanisms for particular scenarios. The research implications and RQ2 are addressed in this section.

7.1. Implications

According to this SLR, it is evident that work-proof-based mechanisms continue to be the most popular (43.3%) from

which 89% used the PoW consensus mechanism, in this category, until August 2022. Voting-based mechanisms come second (29%) from which 50% used the PBFT mechanism, in this category. Followed by capability-based methods, in the last place (27.7%) from which 30% used the PoS, in this category. This indicates that, even though many authors and practitioners believe DPoS-based and PBFT-based mechanisms may be the greatest alternatives to PoW-based mechanisms, their adoption is still far behind PoW. This suggests that within the cryptocurrency industry, PoW continues to be the most widely employed consensus mechanism. Additionally, because Bitcoin and Ethereum were innovators, PoW benefits from being the first to market. While Ethereum is the first platform to use smart contracts in BC, Bitcoin has been the first successful cryptocurrency. Due to its success, other cryptocurrencies may have followed the PoW as their equivalent consensus method. The majority of platforms (e.g., Tron and EOS) appeared after 2016 when Ethereum started using PoW. These platforms use similar tokens of Ethereum based on PoW. This could be the cause of the PoW implementation present in the newest cryptocurrencies. The market distribution shows around $27 Billion in Bitcoin and $14 Billion in Ethereum, dominating all other currencies by a staggering 40% of Bitcoin and 19% of Ethereum [147]. This provides further hints why PoW dominates other mechanisms [22].

Voting-based mechanisms, especially PBFT-based mechanisms, according to this SLR, have shown a big jump in 2022. This indicates that many authors have seen a higher potential in voting mechanisms in FC rather than work-proof-based mechanisms or capability-based mechanisms. This can be due to the nature of FC with limited resources such that voting-based mechanisms may use fewer resources than other mechanisms. However, here we need to remember the high communication overhead since all participant nodes should share in the voting process. Another remarkable note in this SLR is that several new mechanisms are proposed in 2022, which are not even based on traditional mechanisms such as PoW or PoS. This gives another indication that the authors are working towards solving both resource issues related to work-proof-based mechanisms and capability-based mechanisms, and communication overhead issues related to voting-based mechanisms. Accordingly, we are expecting more mechanisms to be proposed in the context of BC-based FC in the near future.

One could ask if the proportion of the consensus mechanisms will change given PoW's supremacy. We think that in the coming years, we will probably see a shifting of proportion. The largest change in this area may be towards the voting-based mechanisms. This SLR demonstrated that the bulk of studies using voting mechanisms were published in 2022, which makes this conclusion quite evident. Additionally, it is thought that capability-based mechanisms like DPoS security will be considerably closer to PoW and far better than any present DPoS mechanism can offer due to their significant emphasis on game-theoretic and economic incentive-based approaches. Specifically, there will be many more validators than are now used by the DPoS mechanisms. It remains to be observed how they will work when used in practical situations, though [22].

7.2. Future directions

According to this SLR, work-proof-based consensus mechanisms have significant drawbacks, particularly the high resources and power requirements as well as low throughput. Moreover, both capability-based mechanisms and voting-based mechanisms have several limitations. In the selection stage, the mining rights are automatically awarded to the node with the largest stake, in capability-based mechanisms, which decreases the wastage of computational resources compared to work-proof-based mechanisms, but it could lead to monopolies. This could fuel the system's trend toward centralization, giving malevolent intruders an obvious target to attack and jeopardizing the system's security. DPoS, for example, forgoes decentralized characteristics in order to increase system throughput [107].

Voting-based mechanisms, on the other hand, have a large communication overhead. It is clear that all of the mechanisms have valid applications, but there is no optimum mechanism. Moreover, as revealed in this SLR, several mechanisms were introduced to reduce power usage and low throughput in capability-based mechanisms, including optimized PoDL [122], PoL [111], and SmartCoin [103]. Similarly, several voting-based mechanisms were introduced to simplify PBFT or reduce communication overhead, such as simplified PBFT [106], SSC [117], and LPoP [97]. However, these proposals are still in the development phase and have not been tested in comparison to other BC consensus mechanisms. Table 7 summarizes the challenges and future directions of BC consensus mechanisms in FC applications.

7.3. Threats to validity

For this SLR, the threats-to-validity component of quality evaluation is done by looking at the titles, keywords, abstracts, and full texts of journal and conference articles to determine how relevant they are to the combination of BC and FC. The validity requirements including, internal validity, external validity, construct validity, and conclusion validity were covered in this study, to ensure the validity and transparency of the findings [148]. To enhance internal validity, we established clear inclusion and exclusion criteria, defining the parameters for selecting studies based on publication dates, relevance to the research question, and documenting the databases searched, and search terms. Also, we implemented blind screening by reviewing and selecting studies, independently. We also followed a data extraction protocol for extracting predefined information from each selected study and conducting quality assessments using established tools or checklists to help address potential bias, which enhances both internal validity and conclusion validity [149]. To enhance the external validity of the review, the inclusion of papers included a diverse set of studies, and various characteristics, such as different populations, geographic locations, and periods. To enhance construct validity and conclusion validity, we meticulously aligned the review process with the research

Table 7
BC consensus mechanisms' challenges and future directions in FC applications.

Algorithm	Study	Considers	Future directions
PoW	[47,84,99, 109]	• High power consumption • Centralized miners • Uselessness computation, forking, unfair incentivization • Publicly accessible which reduce privacy • Building blocks is challenging	• Lightweight PoW to decrease the cost of service and data storage • Using smart contracts to enhance adaptability • Identifying how to reduce the computational resource requirements for BC applications in mobile networks
DPPoW	[59]	• Although it improves resource authentication, it still has problems with high computing requirements	• To realize the performance measurements, real-time sensor devices with a range of capabilities should be tested
ePow	[82,96]	• Although ePoW improves data authentication and protects against data poisoning attacks, it carries over PoW's high power consumption and unfair incentive problems	• Experiments with real public BC networks are required
Groupchain	[50]	• Higher latency than PoW	• Enhancement of the throughput time and incentive mechanism • Experiments with real public BC networks are required
PoS	[40,80]	• Suitable for the consortium in certain scenarios • Vulnerable to nothing-at-stake issue • Building blocks is challenging	• Work is needed to create lightweight versions of this method and address the no-stake issue
DPoS	[102,106, 114,136]	• Forking issues and unfair incentive mechanism • Centralization and high energy use • Difficulty to select the ideal delegates for block development	• DPoS has some drawbacks similar to PoS, such as that only participants can get the block incentive, which significantly reduces the liquidity of coins • Enhance optimization and the stability of the algorithm
PPoS	[79]	• Despite its high throughput and scalability, it is used in consortiums under certain situations	• To realize the performance measurements, real-time sensor devices with a range of capabilities should be tested
Improved DPoS	[114]	• The complexity of the computations will rise as a result of using the mechanism	• Reduce calculating complexity while maintaining accuracy as a priority
PoET	[40]	• Total reliance on Intel rather than any other third party • Specialized hardware is required	• Involves using certain SGX hardware from Intel which exercise of dominating authority results in a more centralized BC
PoA	[39,40]	• Excessive energy use • Centralized authority and validators need to confirm their real identities • Used only for permissioned BCs	• To realize the performance measurements, real-time sensor devices with a range of capabilities should be tested
PoSer	[69]	• Built based on PoW and PoS so it inherits their challenges • Security level depends on the legitimate user's control level	• Identifying how to reduce the computational resource requirements for BC applications
PoD	[60]	• It is similar to PoS in terms of resource usage. When compared to PoS, PoD employs the devices online time of as stakes to reduce the virtual asset computation	• Real-world experiments to determine how to reduce the computing resource needs
Optimized PoDL	[122]	• Communication overhead • Does not have a solution for multitasks • The verification procedure must be repeated numerous times by multiple full nodes	• Practical test for this consensus mechanism as well as comparing it to other mechanisms rather than PoW is required
PoL	[111]	• The speed of block production cannot change in response to BC network capacity	• Enhance the flexibility of the allocation strategy • Experiments in the real world to check its generalizability

(*continued on next page*)

questions. Based on what was learned from the literature, questions were resolved through a series of iterative improvement procedures, cross-checked peer review, and the generation of data according to proof [149].

Table 7 (*continued*).

PoC	[48]	• Decentralization and forking issue • Malware may affect mining activities	• Lightweight PoC to decrease the high resources requirements
SmartCoin	[103]	• Private BC • Does not use any node competition for block submissions	• Assessment in a real-world setting is necessary
BFT	[40,77]	• High complexity • The number of messages increases exponentially • Building blocks is challenging • Suitable for consortium BC	• The consensus approach is yet to be improved
PBFT	[41,78,106, 115]	• Communication overhead • Centralization of computing and high energy use	• Multichain and sidechain to be used to boost model performance • Improve consensus efficiency • Taking into consideration the issue of network access while connecting large-scale devices
DLPBFT	[67]	• Bigger block size compared to PBFT	• Assess the energy usage and take into account the best BC node classification techniques
mPBFT	[87]	• Performance evaluation needs to be compared with other consensus mechanisms	• Assessment in a real-world setting is necessary
AdBFT	[105]	• Performance and security analyses need to be validated	• Mechanism for rewarding task completion and enhancing service quality
SG-PBFT	[41]	• The performance evaluation was only compared to PBFT-based mechanisms	• Improve the miner group selection process to minimize latency time
Simplified PBFT	[106]	• It is more suited to industry fields • The throughput declines as the system's fraction rises • Suitable for consortium BC	• Use extensive node testing in a real-world industrial setting to further enhance its performance in terms of availability and reliability
RAFT	[40,72]	• It is a rigid one-leader protocol • Issues occur in reaching a consensus • Unable to resolve the Byzantine fault-tolerance issue • It is used in permissioned BC	• It is possible to improve data management and decision-making by combining AI with machine learning • No guarantee of data integrity if a node acts maliciously
PoT	[121]	• It works only on a permissioned BC • It requires a minimal threshold of trustworthiness and a minimum number of trustworthy nodes	• Improved versions of PoT to mitigate the effect of forks and computational power wastage
SSC	[117]	• Processing time increases when more nodes are included in the consensus	• Assessment in a real-world setting is necessary • The stability of the miner has an effect on the system; therefore, this issue needs more research
LPoP	[97]	• Security limitations • Block size and latency	• Validate the security level through a wider variety of threats investigation • Enhance and optimize the algorithm when the number of nodes is increased

7.4. Limitations

It is important to acknowledge that this study is limited to the number of selected studies that focus on the specific search parameters defined for this paper. This may inadvertently omit potentially valuable studies that do not precisely align with the predefined criteria. Consequently, the diversity of insights collected could be restricted. Moreover, the reliance on predefined criteria for selecting relevant articles could unintentionally exclude valuable research that presents alternative viewpoints or insights about BC consensus mechanisms. Furthermore, the scope of the SLR was confined to the databases included in the search phase.

While the review covered literature up to August 2022, it is essential to recognize that BC consensus mechanisms continue to evolve daily. Researchers might opt to explore additional databases to uncover further relevant studies. Additionally, the study's conclusions regarding the strengths and limitations of proposed consensus mechanisms are predominantly derived from the assertions made by the authors of the selected studies. These assertions may not necessarily provide a completely accurate reflection of the real-world performance and implications of these mechanisms. Also, the conclusions and recommendations presented in this review are based on the expertise and interpretation of the available literature by the researchers.

It is worth noting that different researchers may arrive at varying conclusions based on their unique perspectives and interpretations. Finally, the classification and decision tree presented in the review are based on the existing literature available at the time and reflect a current assessment. However,

they are subject to potential modifications and updates as new consensus mechanisms emerge in the future. Another limitation for this work is that various BC consensus mechanisms have evolved to better accommodate the unique characteristics of edge computing and IoT devices, in recent years. For example, DHT-based lightchain, Tree-chain, Bullshark, Lattice, and Lachesis consensus mechanisms, among others. However, it is worth noting that there is limited existing literature on these emerging blockchains, primarily because these efforts are still in the development phase.

8. Conclusions

Because of the recent trend toward BC, many FC projects are shifting toward BC-based approaches. The consensus mechanisms necessary to add or verify a new block in the BC are at the heart of the BC. Many research works have recently been made to compare the present consensus methods utilized in BC and propose a new one. In this paper, we conducted a systematic review of available literature on consensus mechanisms using BC-based FC solutions. Based on established criteria, this SLR identified 79 articles for final analysis, as detailed in Section 3.2. This SLR gives a taxonomy of BCs based on their consensus underpinning basis. There were three major themes utilized: work-proof, capability-proof, and voting-based. The three themes were then compared to identify the most suited mechanisms depending on various FC applications. The three themes of consensus mechanisms were analyzed and compared in terms of three dimensions: performance, decentralization, and security.

The findings of this SLR revealed that work-proof mechanisms (e.g., PoW-based mechanisms) remain the dominant theme, but voting-based mechanisms (e.g., PBFT-based mechanisms) have gained a lot of attention in the last two years. The SLR also revealed that PoW is still the most common BC consensus mechanism suggested by many researchers, to provide FC solutions. This SLR also identified several new consensus mechanism proposals. Although the authors have reported that these mechanisms can minimize numerous issues of the previous mechanisms, real-world experiments are required to accurately evaluate the strengths and drawbacks of the proposed consensus in terms of fog node demands. This paper may assist developers in choosing the best mechanism design for their BC application. The paper also highlights the challenges and future research direction for each consensus mechanism, identified in this paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Dave, V. Rastogi, M. Miglani, P. Saharan, N. Goyal, Smart fog-based video surveillance with privacy preservation based on blockchain, Wirel. Pers. Commun. 124 (2) (2022) 1677–1694.

[2] A.S. AlAhmad, H. Kahtan, Y.I. Alzoubi, O. Ali, A. Jaradat, Mobile cloud computing models security issues: A systematic review, J. Netw. Comput. Appl. 190 (2021) 103152.

[3] Z. Ashi, M. Al-Fawa'reh, M. Al-Fayoumi, Fog computing: Security challenges and countermeasures, Int. J. Comput. Appl. 175 (15) (2020) 30–36.

[4] A.R. Nair, S. Tanwar, Fog computing architectures and frameworks for healthcare 4.0, in: S. Tanwar (Ed.), Fog Computing for Healthcare 4.0 Environments. Signals and Communication Technology, Springer, Cham, 2021, pp. 55–78.

[5] R. Arul, Y.D. Al-Otaibi, W.S. Alnumay, U. Tariq, U. Shoaib, M.J. Piran, Multi-modal secure healthcare data dissemination framework using blockchain in IoMT, Pers. Ubiquitous Comput. (2021) 1–13.

[6] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fog computing: Architecture, key technologies, applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42.

[7] H.F. Atlam, R.J. Walters, G.B. Wills, Fog computing and the internet of things: A review, Big Data Cognit. Comput. 2 (2) (2018) 10.

[8] Y.I. Alzoubi, A. Aljaafreh, Blockchain-fog computing integration applications: A systematic review, Cybern. Inf. Technol. 23 (1) (2023) 3–37.

[9] T. Baker, M. Asim, H. Samwini, N. Shamim, M.M. Alani, R. Buyya, A blockchain-based fog-oriented lightweight framework for smart public vehicular transportation systems, Comput. Netw. 203 (2022) 108676.

[10] B. Dammak, M. Turki, S. Cheikhrouhou, M. Baklouti, R. Mars, A. Dhahbi, LoRaChainCare: An IoT architecture integrating blockchain and LoRa network for personal health care data monitoring, Sensors 22 (4) (2022) 1497.

[11] I. Lukić, K. Miličević, M. Köhler, D. Vinko, Possible blockchain solutions according to a smart city digitalization strategy, Appl. Sci. 12 (11) (2022) 5552.

[12] N.S. Khan, M.A. Chishti, Security challenges in fog and IoT, blockchain technology and cell tree solutions: A review, Scalable Comput.: Pract. Exp. 21 (3) (2020) 515–542.

[13] M.A. Ferrag, L. Shu, X. Yang, A. Derhab, L. Maglaras, Security and privacy for green IoT-based agriculture: Review, blockchain solutions, and challenges, IEEE Access 8 (2020) 32031–32053.

[14] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with IoT. Challenges and opportunities, Future Gener. Comput. Syst. 88 (2018) 173–190.

[15] N. Chaudhry, M.M. Yousaf, Consensus algorithms in blockchain: comparative analysis, challenges and opportunities, in: Proceedings of the 12th International Conference on Open Source Systems and Technologies, ICOSST, IEEE, Lahore, Pakistan, 2018, pp. 54–63.

[16] G.-T. Nguyen, K. Kim, A survey about consensus algorithms used in blockchain, J. Inf. Process. Syst. 14 (1) (2018) 101–128.

[17] F. Pelekoudas-Oikonomou, et al., Blockchain-based security mechanisms for IoMT edge networks in iomt-based healthcare monitoring systems, Sensors 22 (7) (2022) 2449.

[18] B. Lashkari, P. Musilek, A comprehensive review of blockchain consensus mechanisms, IEEE Access 9 (2021) 43620–43652.

[19] A. Shahzad, A. Gherbi, K. Zhang, Enabling fog–blockchain computing for autonomous-vehicle-parking system: A solution to reinforce IoT–cloud platform for future smart parking, Sensors 22 (13) (2022) 4849.

[20] Y.I. Alzoubi, V.H. Osmanaj, A. Jaradat, A. Al-Ahmad, Fog computing security and privacy for the internet of thing applications: State-of-the-art, Secur. Priv. 4 (2) (2021) e145.

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

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

[23] J. Nijsse, A. Litchfield, A taxonomy of blockchain consensus methods, Cryptography 4 (4) (2020) 32.

[24] N.C. Gowda, S.S. Manvi, B. Malakreddy, P. Lorenz, BSKM-FC: Blockchain-based secured key management in a fog computing environment, Future Gener. Comput. Syst. 142 (2023) 276–291.
[25] L.A. Ajao, S.T. Apeh, Secure fog computing vulnerability in smart city using machine learning and blockchain technology, Networks 20 (2023) 23–28.
[26] H. Guo, X. Yu, A survey on blockchain technology and its security, Blockchain: Res. Appl. 3 (2) (2022) 100067.
[27] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decent. Bus. Rev. (2008) 21260.
[28] K. Salah, M.H.U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for AI: Review and open research challenges, IEEE Access 7 (2019) 10127–10149.
[29] Y.I. Alzoubi, A. Al-Ahmad, H. Kahtan, A. Jaradat, Internet of things and blockchain integration: Security, privacy, technical, and design challenges, Future Internet 14 (7) (2022) 216.
[30] A. Alkhateeb, C. Catal, G. Kar, A. Mishra, Hybrid blockchain platforms for the internet of things (IoT): A systematic literature review, Sensors 22 (4) (2022) 1304.
[31] H. Honar Pajooh, M. Rashid, F. Alam, S. Demidenko, Hyperledger fabric blockchain for securing the edge internet of things, Sensors 21 (2) (2021) 359.
[32] L. Nkenyereye, B. Adhi Tama, M.K. Shahzad, Y.-H. Choi, Secure and blockchain-based emergency driven message protocol for 5G enabled vehicular edge computing, Sensors 20 (1) (2020) 154.
[33] Y.I. Alzoubi, A. Al-Ahmad, A. Jaradat, V. Osmanaj, Fog computing architecture, benefits, security, and privacy, for the internet of thing applications: An overview, J. Theor. Appl. Inf. Technol. 99 (2) (2021) 436–451.
[34] T. Hewa, M. Ylianttila, M. Liyanage, Survey on blockchain based smart contracts: Applications, opportunities and challenges, J. Netw. Comput. Appl. 177 (2021) 102857.
[35] H.L. Cech, M. Großmann, U.R. Krieger, A fog computing architecture to share sensor data by means of blockchain functionality, in: Proceedings of the 2019 IEEE International Conference on Fog Computing, ICFC, IEEE, Prague, Czech Republic, 2019, pp. 31–40.
[36] Y.D. Al-Otaibi, K-nearest neighbour-based smart contract for internet of medical things security using blockchain, Comput. Electr. Eng. 101 (2022) 108129.
[37] C. Fang, Y. Guo, J. Ma, H. Xie, Y. Wang, A privacy-preserving and verifiable federated learning method based on blockchain, Comput. Commun. 186 (2022) 1–11.
[38] P. Kumar, R. Kumar, G.P. Gupta, R. Tripathi, A distributed framework for detecting ddos attacks in smart contract-based blockchain-IoT systems by leveraging fog computing, Trans. Emerg. Telecommun. Technol. 32 (6) (2021) e4112.
[39] X. Fu, H. Wang, P. Shi, A survey of blockchain consensus algorithms: mechanism, design and applications, Sci. China Inf. Sci. 64 (2) (2021) 1–15.
[40] L. Ismail, H. Materwala, A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions, Symmetry 11 (10) (2019) 1198.
[41] G. Xu, et al., SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent internet of vehicles, J. Parallel Distrib. Comput. 164 (2022) 1–11.
[42] Y.I. Alzoubi, A.Q. Gill, A. Al-Ani, Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. 53 (1) (2016) 22–37.
[43] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, EBSE Technical Report, EBSE-2007-01, 2007.
[44] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Inf. Softw. Technol. 64 (2015) 1–18.
[45] J. Gao, et al., A blockchain-SDN-enabled internet of vehicles environment for fog computing and 5G networks, IEEE Internet Things J. 7 (5) (2019) 4278–4291.
[46] Y. Jiao, P. Wang, D. Niyato, K. Suankaewmanee, Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks, IEEE Trans. Parallel Distrib. Syst. 30 (9) (2019) 1975–1989.
[47] G. Kumar, R. Saha, M.K. Rai, R. Thomas, T.-H. Kim, Proof-of-work consensus approach in blockchain technology for cloud and fog computing using maximization-factorization statistics, IEEE Internet Things J. 6 (4) (2019) 6835–6842.
[48] A. Lakhan, M. Ahmad, M. Bilal, A. Jolfaei, R.M. Mehmood, Mobility aware blockchain enabled offloading and scheduling in vehicular fog cloud computing, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 4212–4223.
[49] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C.P.A. Ogah, Z. Sun, Blockchain-based dynamic key management for heterogeneous intelligent transportation systems, IEEE Internet Things J. 4 (6) (2017) 1832–1843.
[50] K. Lei, M. Du, J. Huang, T. Jin, Groupchain: Towards a scalable public blockchain in fog computing of IoT services computing, IEEE Trans. Serv. Comput. 13 (2) (2020) 252–262.
[51] H. Liu, Y. Zhang, T. Yang, Blockchain-enabled security in electric vehicles cloud and edge computing, IEEE Netw. 32 (3) (2018) 78–83.
[52] N.C. Luong, Y. Jiao, P. Wang, D. Niyato, D.I. Kim, Z. Han, A machine-learning-based auction for resource trading in fog computing, IEEE Commun. Mag. 58 (3) (2020) 82–88.
[53] Y. Qu, et al., Decentralized privacy using blockchain-enabled federated learning in fog computing, IEEE Internet Things J. 7 (6) (2020) 5171–5183.
[54] V. Sharma, I. You, F. Palmieri, D.N.K. Jayakody, J. Li, Secure and energy-efficient handover in fog networks using blockchain-based DMM, IEEE Commun. Mag. 56 (5) (2018) 22–31.
[55] B. Wu, K. Xu, Q. Li, S. Ren, Z. Liu, Z. Zhang, Toward blockchain-powered trusted collaborative services for edge-centric networks, IEEE Netw. 34 (2) (2020) 30–36.
[56] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, Z. Han, Cloud/fog computing resource management and pricing for blockchain networks, IEEE Internet Things J. 6 (3) (2018) 4585–4600.
[57] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, Z. Han, When mobile blockchain meets edge computing, IEEE Commun. Mag. 56 (8) (2018) 33–39.
[58] L. Yang, M. Li, H. Zhang, H. Ji, M. Xiao, X. Li, Distributed resource management for blockchain in fog-enabled IoT networks, IEEE Internet Things J. 8 (4) (2020) 2330–2341.
[59] T. Chen, L. Zhang, K.-K.R. Choo, R. Zhang, X. Meng, Blockchain based key management scheme in fog-enabled IoT systems, IEEE Internet Things J. 8 (13) (2021) 10766–10778.
[60] H. Li, D. Han, M. Tang, A privacy-preserving charging scheme for electric vehicles using blockchain and fog computing, IEEE Syst. J. 15 (3) (2020) 3189–3200.
[61] M. Li, L. Zhu, X. Lin, Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing, IEEE Internet Things J. 6 (3) (2019) 4573–4584.
[62] Y. Gao, W. Wu, P. Si, Z. Yang, F.R. Yu, B-ReST: Blockchain-enabled resource sharing and transactions in fog computing, IEEE Wirel. Commun. 28 (2) (2021) 172–180.
[63] A.A. Abdellatif, et al., MEdge-chain: Leveraging edge computing and blockchain for efficient medical data exchange, IEEE Internet Things J. 8 (21) (2021) 15762–15775.
[64] Q. Kong, L. Su, M. Ma, Achieving privacy-preserving and verifiable data sharing in vehicular fog with blockchain, IEEE Trans. Intell. Transp. Syst. 22 (8) (2020) 4889–4898.
[65] C. Zhang, L. Zhu, C. Xu, BPAF: Blockchain-enabled reliable and privacy-preserving authentication for fog-based IoT devices, IEEE Consum. Electron. Mag. (2021).
[66] Y. Yao, X. Chang, J. Mišić, V.B. Mišić, L. Li, BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services, IEEE Internet Things J. 6 (2) (2019) 3775–3784.

[67] F. Bai, T. Shen, Z. Yu, K. Zeng, B. Gong, Trustworthy blockchain-empowered collaborative edge computing-as-a-service scheduling and data sharing in the IIoE, IEEE Internet Things J. (2021).
[68] C. Núñez Gómez, B. Caminero, C. Carrión, HIDRA: A distributed blockchain-based architecture for fog/edge computing environments, IEEE Access 9 (2021) 75231–75251.
[69] P.K. Sharma, M.-Y. Chen, J.H. Park, A software defined fog node based distributed blockchain cloud architecture for IoT, IEEE Access 6 (2017) 115–124.
[70] Y. Fan, et al., SBBS: A secure blockchain-based scheme for IoT data credibility in fog environment, IEEE Internet Things J. 8 (11) (2021) 9268–9277.
[71] S. Iqbal, A.W. Malik, A.U. Rahman, R.M. Noor, Blockchain-based reputation management for task offloading in micro-level vehicular fog network, IEEE Access 8 (2020) 52968–52980.
[72] S. Jangirala, A.K. Das, A.V. Vasilakos, Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment, IEEE Trans. Ind. Inform. 16 (11) (2019) 7081–7093.
[73] J. Kang, et al., Blockchain for secure and efficient data sharing in vehicular edge computing and networks, IEEE Internet Things J. 6 (3) (2018) 4660–4670.
[74] Y. He, Y. Wang, C. Qiu, Q. Lin, J. Li, Z. Ming, Blockchain-based edge computing resource allocation in IoT: A deep reinforcement learning approach, IEEE Internet Things J. 8 (4) (2020) 2226–2237.
[75] A.V. Rivera, A. Refaey, E. Hossain, A blockchain framework for secure task sharing in multi-access edge computing, IEEE Netw. 35 (3) (2020) 176–183.
[76] Z. Yang, K. Yang, L. Lei, K. Zheng, V.C. Leung, Blockchain-based decentralized trust management in vehicular networks, IEEE Internet Things J. 6 (2) (2018) 1495–1505.
[77] A. Sheikh, V. Kamuni, A. Urooj, S. Wagh, N. Singh, D. Patel, Secured energy trading using byzantine-based blockchain consensus, IEEE Access 8 (2019) 8554–8571.
[78] L. Yang, M. Li, P. Si, R. Yang, E. Sun, Y. Zhang, Energy-efficient resource allocation for blockchain-enabled industrial internet of things with deep reinforcement learning, IEEE Internet Things J. 8 (4) (2021) 2318–2329.
[79] M. Abdel-Basset, N. Moustafa, H. Hawash, Privacy-preserved cyber-attack detection in industrial edge of things (IEoT): A blockchain-orchestrated federated learning approach, IEEE Trans. Ind. Inform. (2022).
[80] Y. Du, et al., Blockchain-aided edge computing market: Smart contract and consensus mechanisms, IEEE Trans. Mob. Comput. (2022).
[81] W. Guo, Z. Chang, X. Guo, P. Wu, Z. Han, Incentive mechanism for edge computing-based blockchain: A sequential game approach, IEEE Trans. Ind. Inform. (2022).
[82] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, A.N. Islam, M. Shorfuzzaman, Permissioned blockchain and deep-learning for secure and efficient data sharing in industrial healthcare systems, IEEE Trans. Ind. Inform. (2022).
[83] M. Li, et al., Cloud-edge collaborative resource allocation for blockchain-enabled internet of things: A collective reinforcement learning approach, IEEE Internet Things J. (2022).
[84] Y. Liang, Y. Li, J. Guo, Y. Li, Resource competition in blockchain networks under cloud and device enabled participation, IEEE Access 10 (2022) 11979–11993.
[85] S.D. Okegbile, J. Cai, A.S. Alfa, Performance analysis of blockchain-enabled data sharing scheme in cloud–edge computing-based IoT networks, IEEE Internet Things J. (2022).
[86] K.N. Qureshi, G. Jeon, M.M. Hassan, M.R. Hassan, K. Kaur, Blockchain-based privacy-preserving authentication model intelligent transportation systems, IEEE Trans. Intell. Transp. Syst. (2022) 1–9.
[87] L. Vishwakarma, A. Nahar, D. Das, LBSV: Lightweight blockchain security protocol for secure storage and communication in SDN-enabled IoV, IEEE Trans. Veh. Technol. 71 (6) (2022) 5983–5994.
[88] Y. Yang, Z. Liu, Z. Liu, K.Y. Chan, X. Guan, Joint optimization of edge computing resource pricing and wireless caching for blockchain-driven networks, IEEE Trans. Veh. Technol. 71 (6) (2022) 6661–6670.
[89] X. Gu, et al., Using blockchain to enhance the security of fog-assisted crowdsensing systems, in: 28th International Symposium on Industrial Electronics, ISIE, IEEE, Vancouver, BC, Canada, 2019, pp. 1859–1864.
[90] K. Kaur, S. Garg, G. Kaddoum, F. Gagnon, S.H. Ahmed, Blockchain-based lightweight authentication mechanism for vehicular fog infrastructure, in: Proceedings of the 2019 IEEE International Conference on Communications Workshops, ICC Workshops, IEEE, Shanghai, China, 2019, pp. 1–6.
[91] M. Alshehri, B. Panda, A blockchain-encryption-based approach to protect fog federations from rogue nodes, in: Proceedings of the 3rd Cyber Security in Networking Conference, CSNet, IEEE, Quito, Ecuador, 2019, pp. 6–13.
[92] M.H. Ziegler, M. Großmann, U.R. Krieger, Integration of fog computing and blockchain technology using the plasma framework, in: Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency, ICBC, IEEE, Seoul, Korea (South), 2019, pp. 120–123.
[93] S. Shukla, S. Thakur, S. Hussain, J.G. Breslin, S.M. Jameel, Identification and authentication in healthcare internet-of-things using integrated fog computing based blockchain model, Internet Things 15 (2021) 100422.
[94] M.A. Uddin, A. Stranieri, I. Gondal, V. Balasubramanian, Blockchain leveraged decentralized IoT ehealth framework, Internet Things 9 (2020) 100159.
[95] M.S. Eddine, M.A. Ferrag, O. Friha, L. Maglaras, EASBF: An efficient authentication scheme over blockchain for fog computing-enabled internet of vehicles, J. Inf. Secur. Appl. 59 (2021) 102802.
[96] P. Kumar, G.P. Gupta, R. Tripathi, TP2SF: A trustworthy privacy-preserving secured framework for sustainable smart cities by leveraging blockchain and machine learning, J. Syst. Archit. 115 (2021) 101954.
[97] P. Bhattacharya, F. Patel, S. Tanwar, N. Kumar, R. Sharma, MB-MaaS: Mobile blockchain-based mining-as-a-service for IIoT environments, J. Parallel Distrib. Comput. 168 (2022) 1–16.
[98] A.A. Khan, et al., A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment, Comput. Electr. Eng. 102 (2022) 108234.
[99] A. Lakhan, M.A. Mohammed, S. Kadry, S.A. AlQahtani, M.S. Maashi, K.H. Abdulkareem, Federated learning-aware multi-objective modeling and blockchain-enable system for IIoT applications, Comput. Electr. Eng. 100 (2022) 107839.
[100] D. Mohapatra, S.K. Bhoi, K.K. Jena, S.R. Nayak, A. Singh, A blockchain security scheme to support fog-based internet of things, Microprocess. Microsyst. 89 (2022) 104455.
[101] M. Poongodi, et al., A novel secured multi-access edge computing based VANET with neuro fuzzy systems based blockchain framework, Comput. Commun. 192 (2022) (2022) 48–56.
[102] H. Qiu, T. Li, Auction method to prevent bid-rigging strategies in mobile blockchain edge computing resource allocation, Future Gener. Comput. Syst. 128 (2022) 1–15.
[103] L. Vishwakarma, D. Das, SmartCoin: A novel incentive mechanism for vehicles in intelligent transportation system based on consortium blockchain, Veh. Commun. 33 (2022) 100429.
[104] Y. Wan, Y. Qu, L. Gao, Y. Xiang, Privacy-preserving blockchain-enabled federated learning for b5g-driven edge computing, Comput. Netw. 204 (2022) 108671.
[105] W. Wang, et al., Privacy protection federated learning system based on blockchain and edge computing in mobile crowdsourcing, Comput. Netw. 215 (2022) 109206.
[106] L. Yang, W. Zou, J. Wang, Z. Tang, EdgeShare: A blockchain-based edge data-sharing framework for industrial internet of things, Neurocomputing 485 (2022) 219–232.

[107] H. Wang, L. Wang, Z. Zhou, X. Tao, G. Pau, F. Arena, Blockchain-based resource allocation model in fog computing, Appl. Sci. 9 (24) (2019) 5538.
[108] D.-Y. Liao, A federated blockchain approach for fertility preservation and assisted reproduction in smart cities, Smart Cities 5 (2) (2022) 583–607.
[109] Z. Mahmood, V. Jusas, Blockchain-enabled: Multi-layered security federated learning platform for preserving data privacy, Electronics 11 (10) (2022) 1624.
[110] S. Wadhwa, S. Rani, S. Verma, J. Shafi, M. Wozniak, Energy efficient consensus approach of blockchain for IoT networks with edge computing, Sensors 22 (10) (2022) 3733.
[111] X. Zheng, Y. Zhang, F. Yang, F. Xu, Resource allocation on blockchain enabled mobile edge computing system, Electronics 11 (12) (2022) 1869.
[112] Y. Xu, G. Wang, J. Yang, J. Ren, Y. Zhang, C. Zhang, Towards secure network computing services for lightweight clients using blockchain, Wirel. Commun. Mob. Comput. 2018 (2018).
[113] X. Qiu, D. Yao, X. Kang, A. Abulizi, Blockchain and K-means algorithm for edge AI computing, Comput. Intell. Neurosci. 2022 (2022) 1153208.
[114] A. Sasikumar, L. Ravi, K. Kotecha, J.R. Saini, V. Varadarajan, V. Subramaniyaswamy, Sustainable smart industry: A secure and energy efficient consensus mechanism for artificial intelligence enabled industrial internet of things, Comput. Intell. Neurosci. 2022 (2022) 1419360.
[115] Z. Wang, et al., An efficient data sharing scheme for privacy protection based on blockchain and edge intelligence in 6G-VANET, Wirel. Commun. Mob. Comput. 2022 (2022) 18.
[116] A. Bonadio, F. Chiti, R. Fantacci, V. Vespri, An integrated framework for blockchain inspired fog communications and computing in internet of vehicles, J. Ambient Intell. Humaniz. Comput. 11 (2) (2020) 755–762.
[117] M. Singh, G.S. Aujla, R.S. Bali, Derived blockchain architecture for security-conscious data dissemination in edge-envisioned internet of drones ecosystem, Cluster Comput. 25 (2022) 2281–2302.
[118] W. Ou, M. Deng, E. Luo, A decentralized and anonymous data transaction scheme based on blockchain and zero-knowledge proof in vehicle networking (workshop paper), in: X. Wang, H. Gao, M. Iqbal, G. Min (Eds.), Collaborative Computing: Networking, Applications and Worksharing. CollaborateCom 2019, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, Cham, 2019, pp. 712–726.
[119] M. Kong, J. Zhao, X. Sun, Y. Nie, Secure and efficient computing resource management in blockchain-based vehicular fog computing, China Commun. 18 (4) (2021) 115–125.
[120] S. Nadeem, M. Rizwan, F. Ahmad, J. Manzoor, Securing cognitive radio vehicular ad hoc network with fog node based distributed blockchain cloud architecture, Int. J. Adv. Comput. Sci. Appl. 10 (1) (2019) 288–295.
[121] U. Jayasinghe, G.M. Lee, Á. MacDermott, W.S. Rhee, Trustchain: A privacy preserving blockchain with edge computing, Wirel. Commun. Mob. Comput. 2019 (2019).
[122] T. Xiang, H. Zeng, B. Chen, S. Guo, BMIF: Privacy-preserving blockchain-based medical image fusion, ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) (2022).
[123] S.A. Latif, et al., AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems, Comput. Commun. 181 (2022) 274–283.
[124] A. Yazdinejad, R.M. Parizi, A. Dehghantanha, Q. Zhang, K.-K.R. Choo, An energy-efficient SDN controller architecture for IoT networks with blockchain-based security, IEEE Trans. Serv. Comput. 13 (4) (2020) 625–638.
[125] Z. Chen, H. Cui, E. Wu, Y. Li, Y. Xi, Secure distributed data management for fog computing in large-scale IoT application: A blockchain-based solution, in: Proceedings of the 2020 IEEE International Conference on Communications Workshops, ICC Workshops, IEEE, Dublin, Ireland, 2020, pp. 1–6.
[126] R.A. Memon, J.P. Li, M.I. Nazeer, A.N. Khan, J. Ahmed, DualFog-IoT: Additional fog layer for solving blockchain integration problem in internet of things, IEEE Access 7 (2019) 169073–169093.
[127] G. Sun, M. Dai, J. Sun, H. Yu, Voting-based decentralized consensus design for improving the efficiency and security of consortium blockchain, IEEE Internet Things J. 8 (8) (2020) 6257–6272.
[128] E.N. Lallas, A. Xenakis, G. Stamoulis, A generic framework for a peer to peer blockchain based fog architecture in industrial automation, in: Proceedings of the 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM, IEEE, Piraeus, Greece, 2019, pp. 1–5.
[129] S. Ismail, et al., Edge IoT-cloud framework based on blockchain, in: Proceedings of the 2nd International Conference on Computer and Information Sciences, ICCIS, IEEE, Sakaka, Saudi Arabia, 2020, pp. 1–7.
[130] A. Seitz, D. Henze, D. Miehle, B. Bruegge, J. Nickles, M. Sauer, Fog computing as enabler for blockchain-based IIoT app marketplaces-a case study, in: 5th International Conference on Internet of Things: Systems, Management and Security, IEEE, Valencia, Spain, 2018, pp. 182–188.
[131] M. Torky, E. Nabil, W. Said, Proof of credibility: A blockchain approach for detecting and blocking fake news in social networks, Int. J. Adv. Comput. Sci. Appl. 10 (12) (2019) 321–327.
[132] C. Pahl, N. El Ioini, S. Helmer, A decision framework for blockchain platforms for IoT and edge computing, in: Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security, IoTBDS, Scite Press, 2018, pp. 105–113.
[133] Y. Qi, M.S. Hossain, J. Nie, X. Li, Privacy-preserving blockchain-based federated learning for traffic flow prediction, Future Gener. Comput. Syst. 117 (2021) 328–337.
[134] A.H. Alkhazaali, A. Oğuz, Lightweight fog based solution for privacy-preserving in IoT using blockchain, in: Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications, HORA, IEEE, Ankara, Turkey, 2020, pp. 1–10.
[135] L.P. Ledwaba, G.P. Hancke, A. Mitrokotsa, S.J. Isaac, A delegated proof of proximity scheme for industrial internet of things consensus, in: 46th Annual Conference of the IEEE Industrial Electronics Society, IEEE, Singapore, 2020, pp. 4441–4446.
[136] S. Zhang, J.-H. Lee, Analysis of the main consensus protocols of blockchain, ICT Express 6 (2) (2020) 93–97.
[137] S. Khezr, A. Yassine, R. Benlamri, Towards a secure and dependable IoT data monetization using blockchain and fog computing, Cluster Comput. 26 (2) (2023) 1551–1564.
[138] S. Namane, M. Ahmim, A. Kondoro, I.B. Dhaou, Blockchain-based authentication scheme for collaborative traffic light systems using fog computing, Electronics 12 (2) (2023) 431.
[139] Z. Guo, G. Wang, G. Zhang, Y. Li, J. Ni, A multi-factor combined data sharing scheme for vehicular fog computing using blockchain, IEEE Internet Things J. (2023) http://dx.doi.org/10.1109/JIOT.2023.3282672.
[140] M. Sánchez-de la Rosa, C. Núñez Gómez, M.B. Caminero, C. Carrión, Exploring the use of blockchain in resource-constrained fog computing environments, Softw. - Pract. Exp. 53 (4) (2023) 971–987.
[141] C. Núñez Gómez, C. Carrión, B. Caminero, F.M. Delicado, S-HIDRA: A blockchain and SDN domain-based architecture to orchestrate fog computing environments, Comput. Netw. 221 (2023) 109512.
[142] Y.I. Alzoubi, A. Alahmad, H. Kahtan, Blockchain technology as a fog computing security and privacy solution: An overview, Comput. Commun. 182 (2022) 129–152.
[143] S. Desai, T. Vyas, V. Jambekar, Security and privacy issues in fog computing for healthcare 4.0, in: S. Tanwar (Ed.), Fog Computing for Healthcare 4.0 Environments. Signals and Communication Technology, Springer, Cham, 2021, pp. 291–314.

[144] C. Sathish, C.Y. Rubavathi, A survey on blockchain mechanisms (BCM) based on internet of things (IoT) applications, Multimedia Tools Appl. 81 (23) (2022) 33419–33458.

[145] S. Samanta, A. Sarkar, A. Sharma, O. Geman, Security and challenges for blockchain integrated fog-enabled IOT applications, in: R.R. Rout, S.K. Ghosh, P.K. Jana, A.K. Tripathy, J.P. Sahoo, K. Li (Eds.), Advances in Distributed Computing and Machine Learning: Proceedings of ICADCML 2022, Vol. 427, Springer, Singapore, 2022, pp. 13–24.

[146] S.W. Turner, M. Karakus, E. Guler, S. Uludag, A promising integration of SDN and blockchain for IoT networks: A survey, IEEE Access 11 (2023) 29800–29822.

[147] CoinMarketCap, Total Cryptocurrency Market Cap [Online].

[148] F.N. Colakoglu, A. Yazici, A. Mishra, Software product quality metrics: A systematic mapping study, IEEE Access 9 (2021) 44647–44670.

[149] G. Dogan, F.P. Akbulut, C. Catal, A. Mishra, Stress detection using experience sampling: A systematic mapping study, Int. J. Environ. Res. Public Health 19 (9) (2022) 5693.