IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 February 2024, accepted 22 February 2024, date of publication 29 February 2024, date of current version 8 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371901

RESEARCH ARTICLE

Mobile Smart Contracts: Exploring Scalability Challenges and Consensus Mechanisms

VIPIN DEVAL[1,2], VIMAL KUMAR DWIVEDI[3], ABHISHEK DIXIT[1,4], ALEX NORTA[5], (Senior Member, IEEE), SYED ATTIQUE SHAH[6], (Senior Member, IEEE), RAHUL SHARMA[7,8], AND DIRK DRAHEIM[7], (Member, IEEE)

[1]Department of Software Science, Tallinn University of Technology, 12618 Tallinn, Estonia
[2]KIET Group of Institutions, Delhi-NCR, Ghaziabad 201206, India
[3]School of Computing, Engineering and Digital Technologies, Teesside University, TS1 3BX Middlesbrough, U.K.
[4]Department of CSE (AI & ML), Vardhaman College of Engineering, Hyderabad 501218, India
[5]Baltic Film, Media and Arts School, Tallinn University, 10120 Tallinn, Estonia
[6]School of Computing and Digital Technology, Birmingham City University, B4 7RQ Birmingham, U.K.
[7]Information Systems Group, Tallinn University of Technology, 12618 Tallinn, Estonia
[8]Ajay Kumar Garg Engineering College, Ghaziabad 201009, India

Corresponding author: Vipin Deval (vipin.deval@taltech.ee)

This work was supported by the Estonian ‘‘Personal Research Funding: Team Grant (PRG)’’ under Project PRG1641.

ABSTRACT Mobile smart contracts (MSCs) are essential to facilitate quick, safe, and decentralized transactions on mobile blockchain networks. Scalable blockchain solutions facilitate the establishment of a mobile blockchain ecosystem characterized by enhanced resilience and adaptability. This encourages an increase in the number of users and, thus, spreads the adoption of blockchain technology in the mobile domain. With the inception of blockchain technology, a wide range of applications use smart contracts due to their high customizability. However, problems with scalability and resource-intensive consensus procedures prevent their general use. Therefore, this work seeks to identify and analyze these constraints by conducting a systematic survey using Kitchenham's guidelines for available scalable blockchains and consensus methods. Out of a preliminary pool of 2,073 publications, our study, which consists of 25 selected studies, identifies 12 consensus mechanisms and 13 scalable blockchain systems. Our investigation shows that, despite the wide range of techniques, no blockchain solution provides the scalability and lightweight operating requirements to implement smart contracts on mobile devices. This realization draws attention to a significant gap in academic and industry-driven blockchain research that may have implications for creating MSCs. Our findings encourage academics to explore scalable and energy-efficient blockchain technology, targeting creating more approachable smart contracts designed with mobile devices in mind.

INDEX TERMS Blockchain, distributed ledger technology, smart contract, scalability, consensus algorithm, proof-of-stake, proof-of-work, peer-to-peer computing, decentralization.

I. INTRODUCTION

The aim of this study is to examine real-world impacts on Web3 adoption by mobile device users, where Web3 mobile adoption is critical to its success. Recently, Web3 has gained tremendous attention, and following major analysts such as Gartner [1] and Harvard Business Review [2], [3], [4], it will stay with us in the future. The Web3 vision takes blockchain to the next level by making disintermediation ubiquitous – establishing disintermediation not only for basic payments but also for a wide range of financial services, digital identities, data, and business models [5], [6]. As such, Web3 consolidates and integrates the fragmented landscape of specific blockchain visions expressed in the many initial coin offerings (ICOs) that we have seen over the past decade. According to Jin and Parrot, ‘‘Web3 is our chance to make a better Internet’’ [3] by making Web3 ‘‘owned and operated by its users’’ [3]. However, a critical success factor for Web3 to take off is its acceptance by users of mobile devices. According to the International Telecommunications

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Union (ITU), in 2022, "mobile continues to dominate as the platform of choice for online access" [7] and "Internet use is becoming as ubiquitous as mobile phones" [7].

A. MOTIVATION

Likewise, mobile devices will be by far the most important means (and, for many, the only means) to access Web3. Take, as an example, the recent massive uptake [8] of the payment service UPI (Universal Payments Interface)[1] in India, which was only possible due to UPI's mobile-first strategy. Similar (and the Web3 vision is even orders of magnitude larger) can be achieved for Web3 also only via mobile uptake. Therefore, a motivation arises to implement web3 technologies, such as smart contracts for mobile devices, to optimize power consumption, storage requirement, and computational power. A *mobile smart contract* (MSC) is defined as a *smart contract* that is part of a lightweight and scalable blockchain that is suitable to be stored on and executed by mobile devices such as smartphones (Def. 1). Hence, an MSC comprises one or more of specific characteristics, i.e., an optimization for reduced power consumption, minimized storage requirements, and a decreased demand for computational cycles. This design approach stands in contrast to the design of conventional smart contracts, which traditionally rely on robust nodes with ample computing and storage capabilities to validate and execute contractual code. An MSC innovatively leverages a lightweight architecture to enable mobile devices to verify and process smart contract transactions efficiently, thus addressing the unique constraints and opportunities presented by the mobile environment. Furthermore, the motivation that arises for SLR on MSCs can be understood with the help of Figure 3, which means that no SLRs are present in the scientific literature that address the need for MSCs.

Blockchain technology creates a decentralized trust environment by redirecting trust to nodes in a peer-to-peer (P2P) network. Blockchain technology is a distributed database that records event data, called transactions, in blocks after being confirmed by network participants [9]. A smart contract leverages the decentralized power of blockchain technology to address the issue of trustworthiness in conventional contracting systems. Different blockchain platforms are utilized to develop smart contracts, Ethereum being the most common [10], [11]. According to studies [12], [13], [14], 44% of organizations are adopting blockchains. Still, studies also refer to the universal issue of transaction throughput that arises from the deployment of blockchain technologies [15], [16]. Another unresolved issue for smart contract-based organizations is that transactions can be manipulated in networks where the user has the majority of control [17], which we refer to as the "centralization" issue in blockchains [150]. Existing consensus algorithms, such as Proof-of-Work (PoW), Proof-of-Stake (PoS), and others, use a percentage-based value for the consensus power distribution, which is a critical factor in public blockchain centralization [18], [19]. The problem caused by public blockchain centralization includes governance in smart contracts due to block size and specific instances of unilateral decision-making forks in blockchains [20].

Recent literature suggests an increasing interest in developing scalable blockchain solutions. Transaction throughput is a crucial performance metric of blockchains, which refers to the number of transactions verified per second (tps) by network nodes with network latency. Current blockchain systems have limited tps with high network latency, resulting in poor network performance. This is mainly due to the verification and processing of sequential transactions. Furthermore, the current blockchain architecture's large size makes it difficult for mobile devices to process the entire blockchain, which is necessary to verify transactions performed by nodes in a peer-to-peer (P2P) network. Several other approaches are proposed in the literature to achieve scalabilities, such as sharding, sharding with ledger pruning, plasma, a committee-based approach, and state channels or on-off blockchain. These approaches aim to achieve sufficient linearity to the number of participants in the network [11], [18].

Developing a lightweight, predictable, and objective consensus protocol could allow low-powered computing devices, such as smartphones, point-of-sale terminals, and Internet of Things (IoT) devices, to participate equally in the generation and validation of blocks [9], [21]. PoW, the most common consensus protocol, is an inefficient and energy-consuming mechanism that has led to a specialized hashing hardware arms race [22]. Alternative consensus protocols, such as PoS [23], Proof-of-Elapsed-Time (PoET) [24], Practical Byzantine Fault Tolerance (PBFT) [25], and Proof-of-Block-and-Trade (PoBT) [26], have been proposed and implemented. Yet, PoS is based on subjective consensus, PoET requires specialized trusted computing environment components, and PBFT relies on multiple rounds of participant voting, requiring more disk space to store all signatures. PoBT is a unique consensus mechanism and an integration framework for IoT blockchains that reduces the computing time to validate transactions and blocks while reducing the memory requirements for IoT nodes [26].

B. CONTRIBUTION

Given that *scalability* and *consensus mechanism* of blockchains are sine qua non for developing a lightweight and energy-efficient blockchain that requires mobile devices (such as smartphones, etc.) to participate in block validation and consensus. To gain a complete understanding of the existing approaches and protocols suitable for MSCs, we conduct a systematic literature review (SLR) following Kitchenham's guidelines. In this paper, we refer to Kitcheham's guidelines as an SLR methodology. Our initial review reveals a gap in the scientific literature regarding the comprehensive evaluation of scalability and consensus mechanisms. To address this

[1] http://cashlessindia.gov.in/upi.html

gap, we conduct an SLR based on our research questions (RQs) outlined in the introduction. We identified 2,073 papers in the first phase of our SLR, and after further selection, we included 135 primary studies for investigation. We extensively examined approaches and algorithms for scalability and consensus mechanisms in these papers and selected 25 papers for data extraction and analysis. Finally, these 25 papers are categorized into two sets: 13 papers on scalability and 12 on consensus mechanisms. This categorization is conceptually represented in Figure 1.

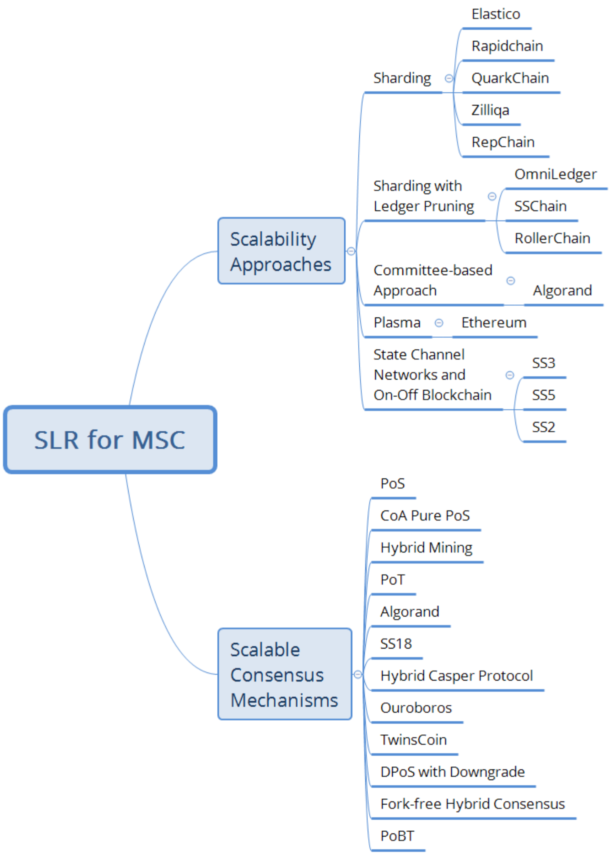


FIGURE 1. Conceptual representation of SLR for MSCs.

Based on the results of our SLR, we discover the gap that prevents the development of MSCs. In the first observation, we find 11 scalable blockchains that achieve scalability through sharding. Still, we find that these blockchains are not lightweight solutions due to inefficient consensus mechanisms and shard connection mechanisms. In the second observation, we find twelve consensus mechanisms suitable for lightweight blockchains, with the PoS consensus algorithm being the best option for MSCs, i.e., lightweight blockchains, since PoS does not necessitate a significant increase in computational power. Still, we investigate the possibility of an oligopoly forming if the major stakeholders significantly influence the network, resulting in centralization. As a result of the lack of an incentive mechanism in the PoS consensus algorithm, the possibility of an oligopoly exists.

In particular, we answer the following research questions:

- *RQ1*: What are the existing approaches and algorithms to enhance the scalability of blockchain and smart contract solutions?
- *RQ2*: What are the different scalable consensus algorithms suitable for MSCs?
- *RQ3*: What are the shortcomings of the algorithms used in RQ1 and RQ2 that hinder the development of MSCs?

The remaining structure of this SLR study is as follows. In Section II, we describe the underlying concepts of blockchain technology and smart contracts to gain knowledge of this disruptive technology. In Section III, we describe the research methodology to conduct this SLR study. Section IV discusses related work that covers all SLRs, reviews, surveys, systematic mapping studies, and comparative studies on blockchain technology and smart contracts. Section V contains the findings that we contribute through this SLR study. In Section VI, we discuss the results of each RQ and describe the advantages, disadvantages, and limitations of the sharding and consensus algorithms. Finally, Section VII concludes the study by summarizing the research findings of this SLR and providing direction for future research.

II. OVERVIEW OF BLOCKCHAIN AND SMART CONTRACTS

Blockchains enable serverless computing in distributed P2P networks, thus ensuring the integrity and security of data in an untrusted environment [27], [28], [17]. This section describes the underlying concepts of blockchain and smart contracts. Section II-A describes blockchain technology, Section II-B describes consensus algorithms, and Section II-C describes smart contracts.

A. BLOCKCHAIN TECHNOLOGY

The first implementation of blockchain was done in 2009 to build a Bitcoin network, thus creating the Bitcoin cryptocurrency [9]. Bitcoin is the first digital currency that became popular due to the distributed, autonomous, and replicated design of blockchain data storage. A blockchain is a distributed, decentralized, immutable, and permanent ledger that stores multiple records in transactions on the P2P network [29], [30]. The notable innovation of the Bitcoin blockchain prevents the double-spending problem in P2P networks without relying on a centralized party. This becomes possible due to the unspent transaction output (UTXO), which is a model that defines the unspent output, which is input for the next transaction on the blockchain network [9], [31], [32], [33], [34]. A blockchain organizes the transactions in the chain of cryptographically linked blocks. The SHA-256 cryptography technique ensures the security of the blocks. This technique enforces the strong integrity of stored data in the ledger by calculating the hash value of each block as its address [35], [17]. Each block contains the information of the previous block, i.e., the hash value of the previous block. Besides, the block has the data, i.e., timestamp, Merkle root of transactions, nonce, etc. The root hash is calculated by adding the hashes of k transactions to a binary tree to form one hash, called the Merkle root [36]. The key feature of this approach is to prevent data manipulation. For any modification in the data, a massive amount of computational power is required because, to do so, an attacker has to change the hashes of each block from the current to the genesis block. The genesis block is the first block created by the blockchain creator, and subsequent blocks are added after the genesis block by the network nodes (miners in the Bitcoin blockchain) [37]. Blocks contain difficulty level; so-called nonce is important for miners to solve to append a block to the latest copy of the blockchain [38], [39], [40]. Miners are called block validators because they collect the transactions from the transaction pool and verify each transaction collected from the latest copy of the ledger. After verification, a miner creates a block of verified transactions and has to calculate the nonce value according to the difficulty level automatically set by the protocol [41], [42]. The calculation of nonce is to prevent the Sybil attack [43], [44], [45] in which an attacker joins the network with different identities [46], [47]. If the miner finds the nonce, then they broadcast the block to the other peers of the network. Peers check the validity of the nonce and the transaction in that block. If everything is correct, then all peers attach the verified block to the latest mined block. After the successful addition of the block to the blockchain, the block miner receives the reward plus the transaction fee in the form of the native cryptocurrency of the particular blockchain [48]. If two or more miners broadcast the verified block simultaneously, then forks occur where the blockchain is divided into two parts at different peers. In this case, the protocol automatically selects the longest chain, and every peer updates his copy, so the fork is resolved [36]. The consensus around uses a mechanism such as PoW [49], [28], PoS [28], [50] to verify and validate the transaction.

B. CONSENSUS MECHANISMS

In the decentralization nature of blockchain technology, the consensus mechanism plays a major role in approving and committing a transaction to a blockchain. Before blockchain, practical Byzantine fault tolerance (PBFT) [51] was introduced in 1999 to tolerate Byzantine faults for state machine replication [52] in a distributed P2P network [53]. The transactions are replicated across multiple peers on the blockchain. To add a new block to the blockchain, all parties must agree on the information in that block. To ensure the validity of the transactions, the peers must verify and confirm the data until the genesis block [54], [55]. For example, the Bitcoin blockchain uses the PoW consensus mechanism to protect networks from malicious nodes. In PoW, the peers who are involved in the mining process are called miners, and for validating the block, they receive a reward as cryptocurrency, e.g., Bitcoin [9], [56], [57], [58]. The miners who compute the nonce first and are declared as a valid block by other miners receive a reward. The block is valid if there is consensus among at least 51% of the peers on the network [24], [59]. PoW is very expensive in terms of power consumption and time. The other blockchain is Qtum,[2] which uses lightweight PoS in terms of energy and time compared to PoW [60], [61]. In the PoS, block generation rights depend on the proportional stake of a node in the network. In addition to PBFT, PoW, and PoS, there are a significant number of consensus mechanisms developed by blockchain researchers according to the discussion in Section V.

C. SMART CONTRACTS

Smart contracts have gained popularity because they enable the use of Turing-complete languages on the protocol layer of the blockchain. A smart contract was introduced in 1996 by Nick Szabo [62]. According to Szabo, contractual terms and conditions can be specified in code. This is now becoming possible due to the distributed, trusted, and immutable nature of blockchain technology [10], [63]. A smart contract is a computer program written in a blockchain-based programming language, e.g., Solidity in Ethereum [64], [65], [149]. A smart contract is different from a traditional contract, as it automatically enforces the contractual terms and conditions [66], [67], [68]. Additionally, due to the immutable and decentralization nature of the blockchain, a smart contract does not involve a third party, resulting in cost-effective and trustless systems [69]. A smart contract, once written and deployed on the blockchain in the form of a transaction, cannot be altered. The Ethereum Virtual Machine (EVM) executes the business logic [61], [60] of the Solidity contract. Due to the imperative nature of Solidity, the contract can be in an infinite loop and may cause network failure. To overcome this problem, Ethereum proposes a concept of gas, which is a cost associated with each transaction that measures the contract expiry. The miners set the gas threshold to execute the transaction. If the gas is lower than the threshold, the miners decline that transaction [61], [70].

Given the importance of the concept of mobile smart contracts for this study, we provide a definition of mobile smart contracts as used throughout the paper in Def. 1.

Definition 1 (Mobile Smart Contract): A *mobile smart contract* (MSC) is defined as a *smart contract* that is part of a lightweight and scalable blockchain which is suitable to be stored on and executed by mobile devices such as smartphones.

III. REVIEW METHODOLOGY

For this review, we selected the systematic review of the literature (SLR) [71] to answer the research questions mentioned in Section III-A. We follow the "Guidelines for performing Systematic Literature Reviews in Software Engineering" [72] proposed by Kitchenham. The SLR provides a summary of selected studies, critically identifies the issues in proposed

[2] https://qtum.org/en

approaches, and the valid evidence available in the existing literature on the scalability and consensus mechanisms of the blockchain. An SLR is a unique method of collecting and ordering evidence in software engineering. We conducted our review in five stages: developing research questions, search approach, study selection, quality assessment of primary studies, and data extraction and analysis. We represent the SLR methodology with the help of Figure 2.

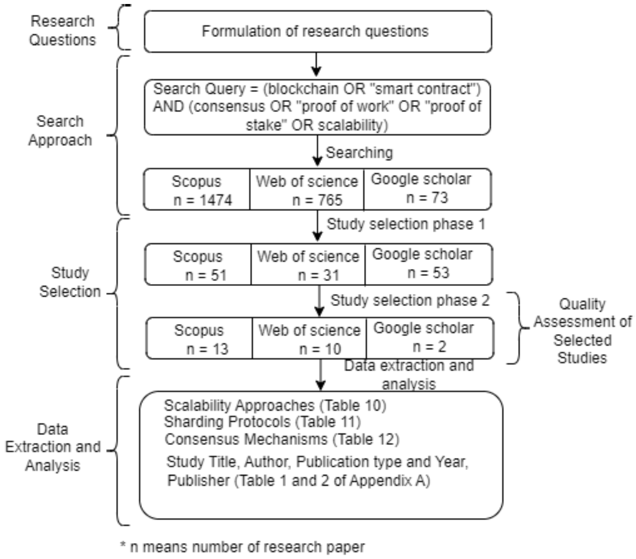

Research Questions
Formulation of research questions
Search Approach
Search Query = (blockchain OR "smart contract") AND (consensus OR "proof of work" OR "proof of stake" OR scalability)
Searching
Scopus n = 1474
Web of science n = 765
Google scholar n = 73
Study selection phase 1
Study Selection
Scopus n = 51
Web of science n = 31
Google scholar n = 53
Study selection phase 2
Quality Assessment of Selected Studies
Scopus n = 13
Web of science n = 10
Google scholar n = 2
Data extraction and analysis
Data Extraction and Analysis
Scalability Approaches (Table 10)
Sharding Protocols (Table 11)
Consensus Mechanisms (Table 12)
Study Title, Author, Publication type and Year, Publisher (Table 1 and 2 of Appendix A)
* n means number of research paper

FIGURE 2. SLR methodology.

A. RESEARCH QUESTIONS

The formulation of research questions is a crucial step in conducting the SLR. In this initial stage, we define the research questions that will guide our investigation of the performance bottlenecks of smart contracts. The identified bottleneck in smart contracts' performance necessitates a thorough examination of the consensus mechanism and scalability. Therefore, our research questions are designed to explore these specific aspects comprehensively. These research questions have been outlined in the introduction section to provide a clear roadmap for our study. In the following sections, we conduct a comprehensive analysis of the existing literature, utilizing these research questions to guide our investigation.

B. SEARCH APPROACH

The search approach is divided into search keywords, scientific databases, and the search process. We identify the search keywords based on the research questions and then generate the search query in Section III-B1. To find relevant articles, we apply the search string in the scientific databases in Section III-B2. Also, we include white papers covering artifacts that are not included in the scientific databases from other sources, such as websites. Finally, Section III-B3 describes the search process.

1) SEARCH KEYWORDS

After checking the keywords of the blockchain and smart contract literature [14], [12], we decided to use 'blockchain' and 'smart contract' as primary search keywords. We also identify secondary search keywords from the research questions, such as consensus, proof-of-work, proof-of-stake, and scalability, which are contained in blockchain research studies. We construct the search query based on the following guidelines to perform the SLR as

(1) Determine search keywords from research questions and the initial literature review.

(2) Decide primary and secondary search keywords from the identified search keywords.

(3) Use the boolean 'AND' and 'OR' operators on the primary and secondary search keywords to form the search query.

Table 1 presents the list of primary and secondary search keywords and the search query for the SLR.

TABLE 1. Search keywords and search query based on the research questions.

Primary Search Keywords	blockchain, smart contract
Secondary Search Keywords	scalability, consensus mechanism, proof-of-work, proof-of-stake
Search Query	(blockchain OR "smart contract") AND (consensus OR proof-of-work OR proof-of-stake OR scalability)

2) SCIENTIFIC DATABASES

The several citation databases, namely Scopus and Web of Science, are appropriate for collecting relevant studies. These citation databases include scientific databases such as ACM Digital Library, IEEE Xplore, Elsevier, Wiley Online Library, MDPI, Springer, etc. We also used Google Scholar to search gray literature (white papers and theses) and for those papers that are not indexed in the citation databases.

3) SEARCH PROCESS

While searching in the citation databases, we find that many papers are indexed in more than one database. Therefore, the chance of collecting duplicate papers is high. First, we search Scopus and Web of Science using our search query and import both lists to the Mendeley[3] library because the latter removes duplicate papers when importing the documents to the library. In this process, a total of 80 duplicate papers are found in the Scopus and Web of Science search results. After this step, we manually searched using Google Scholar and found 160 papers are duplicates because Google Scholar searches all papers in scientific databases that are also included in citation databases. Finally, we find that 240 documents are duplicates in our search result, and Table 2 contains the

[3] https://www.mendeley.com/library/

TABLE 2. List of citation- and scientific databases with the total paper count.

Citation Database or Search Engine	Scientific Databases	Article Count
Scopus (Journal Paper, Conference Paper, and Book Chapter)	ACM Digital Library, IEEE Xplore, Elsevier, Springer, Wiley Online Library, MDPI and SSRN	1475
Web of Science (Journal Paper, Conference Paper, and Book Chapter)	ACM Digital Library, IEEE Xplore, Elsevier, Springer, Wiley Online Library, MDPI and SSRN	765
Google Scholar (research papers, books, reports, thesis, and whitepapers)	ACM Digital Library, IEEE Xplore, Elsevier, Springer, Wiley Online Library, MDPI, and SSRN	73
	Total	2313
	Total (after duplicate removal)	2073

number of relevant papers found from different scientific databases.

C. STUDY SELECTION

After searching for the papers, we performed a two-phase screening to find relevant papers based on the research questions. Table 6 shows the phase-wise selection of the studies.

1) STUDY SELECTION PHASE 1

In this step, we use the Mendeley library to read the title, abstract, and keywords of the papers found in the search result and then assess them based on inclusion and exclusion criteria, as listed in Table 3 and Table 4 respectively. At the end of this step, we find 135 papers (as per Tables 1, 2, and 3 in Appendix A in the supplementary material attached) and check the references of the selected studies for the missing papers in the initial search. However, we did not find any studies that were not on our list of selected studies.

2) STUDY SELECTION PHASE 2

We use NVIVO[4] for full text reading of the articles and qualitative data analysis. First, we import all selected studies in NVIVO from the Mendeley library and apply the quality assessment criteria [72] listed in Table 5 to select the studies for data extraction and analysis. After completion of this phase, we select 13 papers for scalability (as per Table 1 in Appendix A) and 12 papers (as per Table 2 in Appendix A) for consensus mechanisms. Finally, 25 papers were selected from 135 primary studies for data extraction and analysis.

D. QUALITY ASSESSMENT OF SELECTED STUDIES

In this step, we evaluate the quality of the primary studies based on the criteria in Table 5 to prevent biases in the study selection process. To do this, two teams of researchers from this review study independently check two sets of 145 primary studies by answering the questions in Table 5. Each team answers the question with ‘yes’, ‘partly’, or ‘no’ while scoring in numbers 1, 0.5, and 0, respectively. The score from each team is added to derive the average score of each primary paper. Consequently, we select those studies that score 3, or more for data extraction and analysis. At the

TABLE 3. Inclusion criteria for Phase 1 study selection.

Criteria ID	Inclusion Criteria
IC1	Articles that address the scalability of blockchain and smart-contract solutions.
IC2	Articles that discuss the theoretical background on consensus mechanism and scaling blockchain and smart contracts.
IC3	Articles that propose a new consensus algorithm.
IC4	Articles that comprise a new method or algorithm to enhance scalability.
IC5	Articles describing the role of consensus mechanisms and scalability in smart contracts and blockchain.
IC6	Comparative studies on consensus mechanisms.
IC7	Comparative studies of different approaches for scaling blockchains.
IC8	The SLR studies, surveys, systematic mapping studies, and review studies on blockchain and smart contracts.

TABLE 4. Exclusion criteria for phase 1 study selection.

Criteria ID	Exclusion Criteria
EC1	Articles contain use case analysis of blockchain and smart contracts.
EC2	Articles on the security analysis of blockchain and smart contracts.
EC3	Articles on formal verification and correctness analysis of smart contracts.
EC4	Articles presents reviews and surveys on evolution, security, and challenges other than scalability and consensus mechanism.
EC5	Articles on blockchain interoperability services.
EC6	Articles on Language Development for Smart Contracts.
EC7	Articles on the financial and legal perspective of smart contacts and blockchain.

TABLE 5. Quality assessment criteria for phase 2 study selection.

Criteria ID	Quality Criteria
QC1	Are the significance, contribution, and research objectives clearly defined?
QC2	Does the study propose a new approach/method/algorithm to improve scalability and consensus algorithms?
QC3	Does the study clearly describe latency, throughput, fork analysis, and the problems addressed in the study?
QC4	Does the proposed approach contain evaluation or experimental results or simulation results?
QC5	Does the study contain a comparative analysis of results with the existing solution?

[4] https://www.qsrinternational.com/nvivo/nvivo-products

TABLE 6. Step-wise screening of the studies.

Source	Search Result	Study Selection Phase 1	Study Selection Phase 2
Scopus	1475	51	13
Web of Science	765	31	10
Google Scholar	73	53	2
Total	2313	135	25
Total (after duplicate removal)	2073	(Primary studies)	(Selected Studies)

TABLE 7. Quality assessment score of the selected studies.

Selected Study ID	QC1	QC2	QC3	QC4	QC5	Total Score
SS1	0.5	1	0.5	1	0	3
SS2	1	1	1	0.5	0	3.5
SS3	1	1	0	1	0	3
SS4	1	1	1	1	1	5
SS5	1	1	0.5	1	0	3.5
SS6	1	1	1	1	1	5
SS7	1	1	1	1	1	5
SS8	0.5	1	0.5	1	0	3
SS9	1	1	1	1	1	5
SS10	1	1	1	0.5	0	3.5
SS11	1	1	1	0	0	3
SS12	1	1	1	1	0	4
SS13	1	1	0	1	0	3
SS14	1	1	1	0.5	0	3.5
SS15	1	1	1	1	1	5
SS16	1	1	1	1	1	5
SS17	1	1	1	1	0	4
SS18	1	0.5	0.5	0.5	0.5	3
SS19	1	1	1	0.5	0	3.5
SS20	1	1	1	1	0	4
SS21	1	1	1	1	0	4
SS22	1	1	1	1	1	5
SS23	1	1	1	1	0	4
SS24	1	1	1	1	0	4
SS25	1	1	1	1	1	5

TABLE 8. Data extracted from the selected studies.

Extracted Data Element	Reference
Scalability approaches	Refer Table 10
Sharding Algorithms	Refer Table 11
Consensus Mechanisms	Refer Table 12
Study Title	Refer Table 1, 2 (Appendix A)
Author	Refer Table 1, 2 (Appendix A)
Publication Type and Year	Refer Table 1, 2 (Appendix A)
Publisher	Refer Table 1, 2 (Appendix A)

end of this step, we select a total of 25 studies. The scores of each selected study are presented in Table 7 and details of the selected studies, including the title and the journal/conference name, are presented in Tables 1 and 2 in the appendices (attached as supplementary material with this submission).

E. DATA EXTRACTION AND ANALYSIS

The significance of data extraction and analysis is to identify the main contribution of the selected studies. We present the results of this study in Section V and a discussion of the results in Section VI. Table 10 classifies the scalability approaches of the selected studies in Table 1 in Appendix A and from the available scalable approaches, we extract the sharding protocols in Table 11. A comprehensive comparison of scalable consensus algorithms is presented in Table 12, extracted from the selected studies in Table 2 in Appendix A. Finally, Table 8 presents the summary of the data extracted from the 25 selected studies.

IV. RELATED SURVEYS

Most of the literature reviews on blockchain and smart contracts focus on applications, future trends, security, and consensus protocols. The classification of related studies is shown in Table 9. Some studies are specific to the Bitcoin and Ethereum blockchain [50], [73], [74], [75]. In [76], [77], and [78], the authors focus on the scalability of the blockchain, and other studies [79], [80], [81], [82], [83], [84], [85] compare the consensus mechanisms of the blockchain. Still, no studies are present in related work that comprehensively review scalable approaches and consensus algorithms for lightweight blockchains.

Table 9 provides the details of the related work according to the RQs. We classify related studies into four categories such as SLRs, reviews, surveys, and comparative studies. There exist 4 SLRs, 14 reviews, 14 surveys, and 2 comparative studies, respectively, in the table. Also, we have represented related work data with the help of a pie chart in Figure 3. In the SLR, PS33 (as per Table 3 in Appendix A), Shahab et al. [13] review the consensus protocols in public and private distributed ledgers. The study contains reviews of 69 consensus protocols and argues that no consensus protocol is suitable for all contractual business needs of an organization. The study compares three sectors of distributed ledger technologies (DLT), such as public, private, and consortium, and a summary of 69 consensus algorithms. The author of this study does not provide any valid evidence for the 69 consensus protocols. In the PS39 study, Shen et al. [12] selected 159 primary studies from 3827 articles and then 71 articles for data extraction and analysis of primary

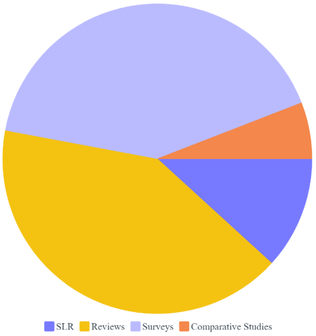


FIGURE 3. A Pie chart for visualizing related work.

studies. The study analyzes the design and prototype present in the 71 papers for the use case of blockchain for cities and the relationship of blockchain with urban sustainability for social, environmental, economic, and governmental growth. In the PS56 study, Casino et al. [14] completed their study on 314 primary studies from the set of 738 articles searched. This study classifies blockchain-based applications, identifies their issues, and comprehensively describes the taxonomy of blockchain applications for future use trends. Yu et al. [86] provide a comprehensive overview of cutting-edge sharding mechanisms, ranging from BFT-based to Nakamoto-based strategies, the latter of which had never been systematized in any existing surveys at the time of writing. This is a study of the benefits and drawbacks of existing intra-consensus-safety solutions, the atomicity of cross-shard transactions, and the overall difficulties and advances proposed by the investigated sharding strategies. They also present a formula to estimate the theoretical top bound of throughput for each sharding strategy considered. A comprehensive comparison is provided based on the results and insights into the features and limits of each available sharding approach. Scalability is a major barrier to adopting blockchain in real-world applications [87]. Scalability issues are addressed by Xie et al. [88] and Hafid et al. [89] from the perspectives of throughput, storage, and networking. PS129 [90] An SLR on scalability issues and challenges of the public blockchain.

The second category reviews architecture, consensus algorithms, blockchain with IoT, and blockchain-based applications. The two studies in this category reviewed specific scalability issues for Bitcoin and Ethereum. Lashkari et al. [91] present a comparative assessment of blockchain as one of the antecedent types of distributed ledger and classification of consensus methods. They also compared consensus algorithms in terms of scalability, finality, adversary tolerance, accessibility, agreement, incentives, centralization, and cost. Three studies investigated the performance of the PoS algorithm, and then four studies reviewed the security and privacy of the blockchain. The two studies lack valid evidence for the issues mentioned above, while none of the studies critically assess the research gaps in blockchain technology. The third category of surveys lists all studies that provide surveys on the design specification of Bitcoin, blockchain taxonomy, and architectures. Some studies provide details of mining strategies [49], [92], long-range attacks on PoS [93], and security vulnerabilities of Ethereum smart contracts [94]. The fourth category presents comparative studies that contain comparisons of security attacks, consensus algorithms, and evaluations of consensus mechanisms in cryptocurrencies. The above data are in Table 9, and the list of studies is in Table 3 in Appendix A.

V. RESULTS

This section presents the result of the SLR answering research questions. The answer to RQ1 is given in Table 10-11, which contains the list of scalability approaches and presents only those approaches that can be suitable for MSCs, respectively. Table 12 answers RQ2 by presenting the details of the scalable consensus algorithms. Finally, the RQ3 answer presents a detailed analysis of the sharding and consensus algorithms to find possible research gaps in the state of the art.

RQ1: What are the existing approaches and algorithms to enhance the scalability of the blockchain and smart contract solutions?

This section synthesizes the knowledge presented in the selected studies and describes the identified scalability approaches. We examine the scalability approaches with their parameters, namely, the name of the smart contract platform, the consensus mechanism used, and the throughput. The parameters are critical to understanding the difference between state-of-the-art scalable smart contract platforms. With these parameters, we establish a relation between these approaches and categorize them into five categories, namely sharding, sharding with ledger pruning, committee-based approach, plasma, and state channel networks, or on-off blockchain (see Table 10).

The consensus mechanism and throughput parameters indicate the motivation behind the development of scalable blockchains. The nature of the scalability approach infers its suitability for the development of lightweight blockchains. With the latter observation, only sharding and sharding with ledger pruning are best suited for MSCs because sharding and related approaches divide the whole blockchain into several parts called shards. A shard could be easily stored and processed on a mobile device. With this motivation, we further investigate the smart contract platforms of sharding and sharding with the ledger pruning approach and identified attributes, namely, sharding/committee type, consensus mechanism, number of nodes, resiliency, latency, throughput, shard size, and cross-shard communication (see Table 11). These attributes are essential, as they provide critical insight for developing lightweight blockchains with enhanced throughput.

A. SCALABILITY APPROACHES

We identify the scalability approaches from the selected studies in Table 10. In this table, the approaches are categorized based on their method of scaling, consensus mechanisms,

TABLE 9. Classification of related work.

Study Type	Study IDs	Contributions of the Study
SLR	PS33 [13]	Applicability and suitability of consensus mechanisms No single consensus algorithm has fulfilled all business needs
	PS39 [12]	Blockchain use case review for the urban sustainability Investigate a relationship between application areas and actual system prototype
	PS56 [14]	The current status and application classifications of various blockchain-based application and open issues in the applications
	PS129 [90]	An SLR about public blockchain scalability issues and challenges.
Reviews	PS49 [95]	Characteristics and performance of consensus mechanisms
	PS47 [96]	Comprehensive overview of blockchain architecture and comaprison of consensus protocols
	PS45 [43]	Overview of blockchain technology including architecture and consensus algorithms
	PS118 [76]	Scalability issues of Ethereum and Bitcoin and recent proposed solutions
	PS53 [28]	Mechanism of smart contracts and combination of blockchain with the internet of things
	PS35 [36]	Review of blockchain security and privacy
	PS29 [94]	Investigate PoS algorithm with performance analysis
	PS110 [73]	Intrepretation of smart contracts on Bitcoin and Ethereum
	PS108 [97]	Systematic investigation of blockchain and cryptocurrency on model, techniques and applications
	PS115 [98]	Review of blockchain and applications based-on blockchain
	PS116 [99]	Overview of blockchain-based smart contracts and their future trends
	PS126 [101]	Investigate blockchain scalability-related technologies in terms of enhancing efficiency and expanding the usefulness of the blockchain system.
	PS128 [91]	An in-depth exploration of the fundamentals of distributed ledger technology and its variants.
	PS131 [85]	Reviews blockchain scalability for improving efficiency.
Surveys	PS51 [102]	A Survey on technological and application perspective of blockchain
	PS52 [103]	A technical survey on the design specification of Bitcoin and its applications
	PS36 [104]	Comparative study of blockchain on taxonomy, architecture, and applications
	PS27 [100]	A survey on requirements, challenges, and future trends of blockchain-based internet of things
	PS31 [49]	A survey on mining strategies and consensus algorithms in blockchain networks
	PS30 [93]	A survey on long-range attacks on PoS blockchains
	PS104 [50]	Highlighted the security vulnerabilities of Ethereum smart contracts.
	PS106 [105]	A survey on various consensus protocols
	PS107 [74]	A survey to provide security and privacy in IoT using blockchain technology
	PS123 [86]	A survey on sharding in blockchain provides theoretical information regarding the throughput for each sharding strategy.
	PS124 [88]	Examine scalability regarding throughput, storage, and network connectivity.
	PS125 [89]	Provides a performance-based assessment of the advantages and disadvantages of the available scalability alternatives (i.e., throughput and latency).
	PS134 [106]	Identify the five fundamental components of a blockchain consensus protocol: block proposal, block validation, information propagation, block finalization, and incentive mechanism. The consensus protocols surveyed are analyzed using the five-component framework and compared with various performance metrics, including throughput and latency.
	PS135 [107]	Examines the available blockchain approaches for scalability and classifies them by level. In addition, they provide a comparison of several approaches and a list of potential solutions to the blockchain scalability issue.
Comparative studies	PS40 [79]	Comparative analysis of consensus algorithms such as PoW, PoS, and DPoS with their applications.
	PS34 [81]	Comparative evaluations conducted on significant consensus mechanisms.

and throughput (in tps). The idea of selecting the consensus mechanism and throughput is to compare which approach achieves significant throughput. Existing algorithms achieve scalability using either a PoW or PBFT consensus mechanism, which is not feasible for mobile devices. We discover that PoS is faster than PoW in consensus finality [21]. With the initial review of the literature, we define for the scalability approach five categories, such as sharding, sharding with ledger pruning, committee-based approach, plasma, and state channel network [108]. Furthermore, we select the sharding algorithms as in Table 11 refined from Table 10. In the following, we present the details of the approaches defined in Table 10.

Sharding is introduced to scale up the distributed databases by partitioning the databases horizontally [109]. Similarly, sharding is applied to blockchains by dividing the P2P network into nodes called a shard. The latter is responsible for processing only the data related to that shard. In blockchains, the sharding method is divided into two types: state sharding and transaction sharding. In state sharding, a shard stores a disjoint part of the blockchain and processes the related transactions. On the contrary, in transaction sharding, each node in a shard stores a full copy of the blockchain and processes the disjoint set of transactions in parallel. Sharding in blockchains has a two-layer architecture, i.e., root-chain and shard-chain networks. Every shard in the network maintains a subchain called a shard chain that stores the data validated by the nodes of that particular shard. After processing a block in each shard, the leader sends the block header to the main chain called the root chain, combines all the data received from each shard, and then appends to the main chain using consensus mechanisms. The root-chain network prevents a double-spending attack even if malicious nodes influence a shard. In the literature,

TABLE 10. List of the scalability approaches with their platforms/study ID and implementation details.

S. No.	Scalability Approach	Smart Contract Platform/ Study ID	Consensus Mechanism	Throughput (in tps[1])	Storage Overhead	Communication Overhead	Implementation Details
1	Sharding	Elastico [110]	PoW[2]and PBFT[3]	Linear[4]	High	High	Shards, Subnets, Parallel Processing
		RapidChain [115]	PoW, Gossiping Protocol and Synchronous Consensus Protocol	7384	Moderate	High	
		QuarkChain [111]	PoW	Linear	Moderate	Moderate	
		Zilliqa [112]	PoW and PBFT	2488	High	High	
		RepChain [114]	Synchronous BFT consensus	6852	Moderate	Moderate	
2	Sharding with Ledger Pruning	OmniLedger Or ByzCoin [116]	PoW and PBFT	2250	Low-Moderate	Moderate	Shards, Pruning, Compact Storage
		SSChain [117]	PoW	6500	Low	Moderate	
		Rollerchain [113]	PoW and PBFT	High[5]	Low-Moderate	Moderate	
3	Committee-based Approach	Algorand [118]	Byzantine Agreement and Verifiable random function	125x of Bitcoin[6]	Moderate	Moderate	Committees, Random Selection, Secure Voting
4	Plasma	Ethereum [119]	PoW	NA[*]	High	High	Child Chains, Main Chain Commitments, Fraud Proofs
5	State Channel Networks and On and Off-Blockchain	SS3 [120]	PoW	NA	Low	Low	State Channels, Locking, Off-chain Transactions
		SS5 [121]	NA	NA	Low	Low	
		SS2 [122]	NA	NA	Low (Off-chain)	Low	

[*] Data is not available in the study.
[1] Number of transactions verified in one second.
[2] Proof-of-Work.
[3] Practical Byzantine Fault Tolerance.
[4] Transaction throughput is near linear to the total computational power of the network.
[5] As compared to the first sharding protocol Elastico.
[6] Currently, the Bitcoin network has 7 tps throughput, so 125x gives 125*7=875 tps.

we find five algorithms, i.e., Elastico [110], QuarkChain [111], Zilliqa [112], Rollerchain [113], RepChain [114] applied transaction sharding while three algorithms, i.e., RapidChain [115], OmniLedger [116], and SSChain [117], apply state sharding. Algorand [118] uses a committee-based approach that requires storing a full blockchain copy by every network node.

In *Sharding with Ledger Pruning* approach, together with sharding pruning, removes the noncritical information in a block while appending that block to the blockchain. By deleting outdated and superfluous data from the blockchain, sharding with ledger pruning further boosts scalability. This not only offers the advantages of sharding but also improves storage effectiveness. The advantage of the latter is to reduce the size of the blockchain in a shard and root chain. With a rapid increase in the size of the blockchain, fewer nodes can store a blockchain and verify transactions, leading to centralization. Therefore, ledger pruning is applied to make blockchains lighter in shard and root chains. OmniLedger [116], SSChain [117], Rollerchain [113] apply to the shard with ledger pruning to create a scalable and efficient blockchain network. OmniLedger and SSChain apply state sharding from the above three algorithms, while Rollerchain uses transaction sharding.

Committee-based Approach approach is based on a new Byzantine Agreement (BA) protocol called BA* that uses Verifiable Random Functions (VRF) to form a committee of all nodes in the network. A committee is a small set of nodes selected from the total nodes of the network to run the consensus protocol and agree on the next block to add to the blockchain. In committee-based techniques, a few selected nodes are chosen, rather than the entire network, to validate transactions. A secure voting method is often used in the validation process, and committee members are frequently chosen randomly. The latter reduces transaction latency and increases transaction throughput. In this approach, the consensus depends on the committee of the weighted users, which are selected based on some fraction (at least 2/3 of the honest users) of a user's money. This approach is the same as the PoS consensus mechanism because the algorithm uses a weighted user and VRF to form a consensus committee. On the contrary, PoS uses a stake to claim block generation rights and random number generation to elect the next validator [118].

TABLE 11. List of sharding-based protocols.

Protocol, Study ID	Sharding/Committee Type	Consensus Mechanism	No. of Nodes	Resiliency	Latency	Throughput (in tps[1])	Shard Size	Cross-shard comm.
Elastico, SS9 [110]	Transaction Sharding	PoW to set up the committee and PBFT in each committee and the shard chain	n=1600	t< $\frac{n}{4}$[#]	800 Sec.	Linear[2]	m=100	Yes
RapidChain, SS7 [115]	State Sharding	PoW, Gossiping and Synchronous Consensus	n=4000	t<n/3	8.7 Sec	7384	m=600	Yes
OmniLedger, SS6 [116]	State Sharding	PoW, PBFT, Combination of RandHound and VRF[3]-based leader election	n=2400	t<n/3	8.7 Sec	2250	m=250	Yes
SSChain, SS4 [117]	State Sharding	PoW at root and shard chain	n=1800	t<n/4	100 ms	6500	m=15	Yes
Rollerchain, SS1 [113]	Transaction Sharding	PoW to join community and PBFT for consensus	NA[*]	t<n/3	300 Sec	High[4]	NA	No
Zilliqa, SS10 [112]	Transaction sharding	PoW to join the network and PBFT at DS[5] committee and shard chain	n=3600	t<n/4	NA	2488	m=800	Yes
Algorand, SS8 [118]	Committee-based	Byzantine Agreement and Verifiable random function	NA	NA	Low[6]	875	NA	Yes
QuarkChain SS11 [111]	Transaction sharding	PoW at root- and shard chain	NA	NA	150 Sec.	NA	NA	Yes
RepChain SS13 [114]	State Sharding	Synchronous BFT consensus	n=1800	t<n/3	58.2 Sec.	6852	m=225	-

[*] Data is not available in the study.
[#] t and n are the total number of malicious nodes and the total number of network nodes.
[1] Number of transactions verified in a second.
[2] Transaction throughput is close to linear to the computational power of the entire network.
[3] Verifiable Random Functions.
[4] Compared to the first Elastico sharding protocol.
[5] Directory Services.
[6] Compared to Elastico.

Plasma is a framework for executing scalable smart contracts up to a significant amount of state change in a second by allowing the blockchain to run a significant amount of decentralized applications. Plasma comprises two components: MapReduce functions on all blockchain computations and a PoS consensus mechanism on top of the root chain (e.g., Ethereum). In this framework, more than one plasma blockchain (child chain) is connected with the root chain while sending the block headers to the root chain, called proof of fraud. Child chains, root-chain commitments, and evidence of fraud enable the development of child Ethereum blockchains connected to the main Ethereum blockchain. Transactions can now be handled on the child chains rather than the main chain, which reduces the burden on the main chain. Ethereum works only on plasma and sharding to increase transaction throughput in the Ethereum blockchain [119].

State Channel Networks [120], [121], is the modified approach of off-chain protocols [122] that include payment channels and networks. Off-chain protocols execute massive transactions without costly interaction with the blockchain and update the current state on the blockchain after off-chain execution [122]. The state channel network allows the execution of smart contracts and cryptocurrency since off-channel protocols only allow the execution of payments. Participants can perform off-chain transactions and use two-way communication channels. The blockchain is then updated with the status of these transactions. This method reduces the burden on the blockchain network by processing multiple transactions off-chain before requiring an on-chain transaction. In our study, we select three studies that include state channel networks and off-blockchain smart contract enforcement. State channel networks significantly improve off-chain protocols, as this technique allows channel virtualization between two contracting parties on the ledger. In channel virtualization, two parties can open a virtual channel with an intermediary to process the transactions between them. State-channel networks reduce transaction latency and improve scalability because the transaction is completed through an intermediary rather than waiting for confirmations from the network nodes as in Bitcoin and Ethereum [122], [120]. Studies of this category do not present statistical data to claim enhanced scalability; rather, they discuss transaction throughput and latency based on state channels. With our observation of these approaches, we argue that the above approaches are irrelevant for mobile devices, as storing and processing the whole blockchain is necessary for this category.

Scalability techniques have a significant real-world influence on many different industries when applied to mobile blockchain technology. The following are some significant ways that scalability influences the application of MSCs:

- *Mobile Payments:* Mobile blockchain applications, such as digital wallets and payment apps, have improved scalability, allowing them to conduct transactions more quickly and effectively. As a result, blockchain-based mobile payments may become more widely used, offering a safe and decentralized alternative to conventional banking and payment systems [8].
- *Decentralized Social Media:* Scalable mobile blockchain technology impacts the growth of decentralized social media platforms. A more private and censorship-resistant social network experience on mobile devices may be possible due to its ability to handle numerous transactions and interactions in a decentralized fashion [123].
- *Gaming:* The scalability of blockchain on mobile platforms can benefit the gaming sector. It enables decentralized, transparent, and fraud-resistant gaming economies and facilitates the production and exchange of in-game assets [124].
- *Chain Management:* Blockchain-enabled mobile applications can support transparent and unchangeable product tracking. These applications are suitable for larger and more complex supply networks due to improvements in scalability, which enable them to handle massive volumes of data generated in the supply chain [125].
- *Decentralized Identity and Verification Systems:* Decentralized identity applications can be supported by scalable blockchain systems on mobile devices. These provide individuals with authority over their data and a safe mechanism for verifying identities, which can be helpful in some situations, including voting and access control [126].
- *Internet of Things (IoT):* Scalable blockchain technologies can help IoT devices, many of which are mobile or edge devices. These systems can facilitate transactions and secure communication between many IoT devices [127].
- *Healthcare:* A blockchain can safely store and exchange patient records and other health information. The potential for complete mobile health applications that use blockchain to enhance data security, privacy, and interoperability is possible if this can be done at scale [87].

These and other applications can be made more useful and efficient by increasing the scalability of blockchain systems on mobile devices. Scalability is essential to ensuring that blockchain technology can be used to its full potential on mobile platforms, which are increasingly important in many aspects of our digital lives.

B. SHARDING PROTOCOLS

In this section, we refine our results from the first part of RQ1 to discover scalable protocols. With a detailed analysis of scalability approaches, we identify that sharding, sharding with ledger pruning, and committee-based approach are favorable solutions to initiate the development of smart contracts for mobile devices. Table 11 details all the sharding and related protocols with their performance parameters, such as the consensus mechanism, number of nodes, resiliency, latency, transaction performance, shard size, and cross-shard communication. The first blockchain sharding protocol was Elastico [110], designed by Loi Luu in 2016, and since then, there has been a significant development in the sharding protocols to design scalable blockchains. We select nine sharding and related protocols to answer the second part of RQ1. Rollerchain and QuarkChain do not present statistical data to compare with other algorithms, as shown in Table 11. Furthermore, we have depicted the performance of sharding-based protocols based on throughput, latency, and number of nodes in Figure 4. Still, Rollerchain claims high transaction throughput compared to Elastico. The RapidChain claims the highest transaction throughput, i.e., 7384 tps with 8.7-sec transaction latency. RepChain is another blockchain algorithm based on sharding that is reputation-based, fast, and secure. RepChain achieves a transaction throughput of 6852 tps with 58.2 s of user-perceived latency. Elastico partitions the network into smaller groups (shards); each shard is capable of processing transactions in parallel and runs BFT consensus in each shard. Elastico runs PoW to elect the committee of nodes for each shard, and then the committee of nodes runs BFT to agree on a set of transactions from the shards. Therefore, Elastico maintains a rootchain that contains all transactions and a childchain that contains a particular set of transactions. Elastico includes the reconfiguration of shards to protect against Byzantine adversaries in each epoch. OmniLedger applies state sharding, which divides the ledger to be stored on different nodes, thus reducing storage needs. It runs PoW to set the initial identity of the nodes and Byzcoin (a variant of PBFT) within each shard. OmniLedger uses the Atomix commit protocol to maintain consistency across shards and partially applies ledger pruning to maintain ledger size. RapidChain applies state sharding to partition the blockchain state into shards and runs a unique gossiping protocol for message propagation across shards and synchronous BFT consensus within shards to agree on a set of transactions. RapidChain ensures security and achieves high transaction throughput due to the synchronous BFT protocol. Zilliqa applies transaction sharding, in which the whole network is divided into shards, and each shard processes a fraction of transactions. Zilliqa runs PoW to establish node identity and network sharding and PBFT for the consensus on transactions

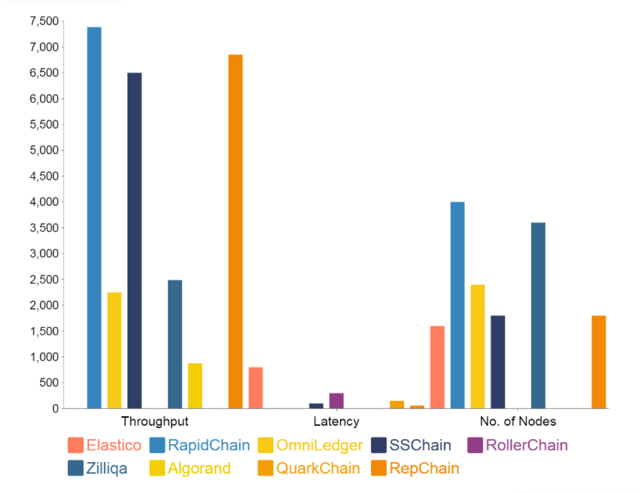


FIGURE 4. Performance of sharding-based protocols.

within shards. RepChain partitions the network into shards based on the reputation score of each node to improve security and efficiency. It runs synchronous BFT consensus to agree within shards. It prioritizes transaction processing based on the node's reputation and identifies malicious nodes with no reputation. Algorand is a committee-based protocol in which they form a committee of nodes using a random verifiable function and a variant of pure PoS and run Byzantine agreement within the committee to reach a consensus.

RQ2: What are the different scalable consensus algorithms suitable for MSCs?

C. SCALABLE CONSENSUS MECHANISMS

In this section, we discover the scalable consensus mechanisms with parameters such as block generation rights, block generation time, scalability, energy efficiency, the chance of fork, etc. in Table 12 and performance of these consensus algorithms is shown in Figure 5. (Note: The actual data in Table 12 is in terms of "High, Low and Moderate"; however, in the figure, we represent data in terms of numbers as 3 for high, 2 for moderate, and 3 for low.) The idea of selecting the performance parameters to compare the consensus algorithms is based on the well-known PoW consensus algorithm. PoW consensus is not a scalable consensus algorithm as in the Bitcoin blockchain; It takes approximately 10 minutes to validate a block, and six block confirmations are required to approve a transaction, that is, 1 hour [9], [103]. Therefore, the Bitcoin blockchain has about 6-7 transactions throughput [137], and the Ethereum blockchain also uses PoW, which has approximately 14 tps [61]. The above five performance parameters select the consensus mechanism for scalable blockchains. A total of 12 consensus algorithms are chosen, of which eight algorithms use PoS with a combination of other consensus algorithms, such as PoW, flexible proof-of-activity (PoA), etc., to make hybrid consensus. The flexible chains of activity (CoA) is a fork-free hybrid consensus with tunable PoW and PoS parameters and an upgradable version of Bentov's PoA [138]. The study SS14 discusses CoA with pure PoS to address rational forks in the blockchain. Another version of the PoS algorithm is Delegated Proof-of-Stake (DPoS), which confines the mining nodes by selecting the delegates for the validation process [100], [139]. The SS16 study contains the Proof of Trust (PoT) consensus mechanism that does not use PoW or PoS, and the trust value of the participant and is based on the trust value of the participant [129]. Assuming that honest miners hold the majority of mining power in the Bitcoin system, the blockchain meets several crucial security features (which are critical for blockchain-based applications).

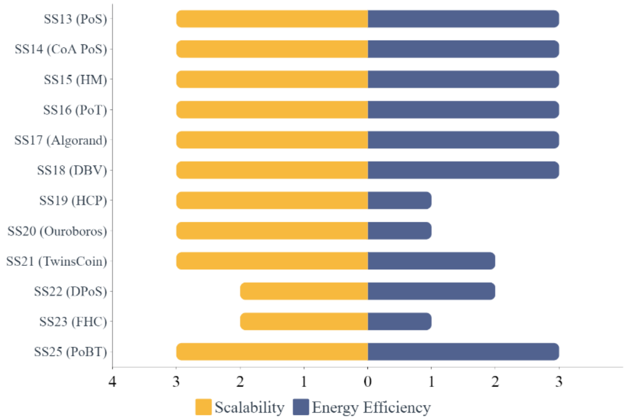


FIGURE 5. Performance of scalable consensus algorithms.

In contrast, if the assumption of "the honest majority of computer power" fails, i.e., the Bitcoin blockchain will be untrustworthy if the adversary controls the system's computing resources. The proof-of-space (PoC) concept has been specially researched in the context of blockchain protocols and is an alternative to PoW. A "prover" in a PoC setting aims to demonstrate the use of space (storage/memory). Like a PoW, this uses a physical resource but is less energy-demanding over time. The proof of space-time (PoST) is a related notion [82]. Still, each scenario requires a costly physical resource, either storage or computational power. Additionally, malicious nodes may exist in any blockchain. Malicious nodes break the trusted consensus mechanism, alter transaction information, generate congestion, and disrupt the regular operation of the network. As a result, a blockchain system can become unreliable, unsecured, and inefficient. To address the above issues, the downgraded DPoS presents an improved consensus that combines PoW's notion of improving fairness with the idea of DPoS of reducing resource consumption and improving the efficiency of the consensus of the blockchain system [135]. The following are the essential features of our revised consensus algorithm: The blockchain selects a set of nodes with enough computational power to participate in the next election and block generation using the PoW algorithm; each node has only one vote for randomly voting, reducing the

TABLE 12. Scalable consensus protocols with performance parameters.

Study ID	Consensus Algorithm	Block Generation Rights	Execution Time	Scalability	Energy Efficiency	Communication Overhead	Chance of Fork	Problems Addressed
SS13 [23]	PoS	Based on the stakes at a node	Low*	High*	High*	Low	Moderate*	- Nothing-at-stake problem
SS14 [21]	CoA Pure PoS and Dense-CoA Pure PoS	Based on unspent transaction output coins	NA	High	High	Moderate	Low	- Rational Forks and - Bribe Attack
Hybrid Mining SS15 [128]	Real-world problem solution with Hashcash and PoW	Based on the solution of the real-world hard computational	NA	High	High	High	Low	- Solutions of real-world NP-complete problems
Proof-of-Trust SS16 [129]	Proof-of-Trust (PoT) consensus using RAFT Leader Election	Service participant's trust value	NA	High	High	Moderate	Low	- Improves scalability in the existing protocols - Removes consensus partiality and unfairness
Algorand SS17 [130]	Graded Consensus Protocol	Based on the money of a user	Low	High	High	Low	Very Low	- Solves the problems of classical Byzantine agreement
SS18 [131]	Dynamic blind voting	Based on voting results in the consensus	Low	High	High	Moderate	NA	- Low transaction throughput - block generation rate - Tampered block broadcast
Hybrid Casper Protocol SS19 [132]	Casper FFG protocol	Depositing the tokens on the blockchain	Low	High	Low	High	Low	- Reducing energy consumption - Nothing-at-stake problem - long-range attack
Ouroboros SS20 [133]	PoS	Stakes at a node	Low	High	Low	Low	Low	- Grinding attacks - Randomized leader election - Persistence and liveness guarantees
TwinsCoin SS21 [134]	Pow and PoS	Based on the computing power and stake	Moderate	High	Moderate	Moderate	Low	- Resiliency against 51%attack
DPoS with Downgrade SS22 [135]	DDPoS	Computation capacity and stake	Moderate	Moderate	Moderate	Moderate	Low	- Achieve high efficiency - Reduce centralization and -Prevent the malicious nodes with downgrade mechanism
Fork-free Hybrid Consensus SS23 [136]	Flexible: Proof-of-Activity	PoW power and PoS capability	Low	Moderate	Low	Moderate	Low	- Fork resolution
PoBT, SS25 [26]	Proof of block and trade	Based on block generated and traded	NA	High	High	Low	NA	-Scalable solution for IoT

* With respect to the PoW consensus mechanism.

impact of stakes on consensus node election; a downgrading mechanism is used to quickly downgrade malicious nodes and upgrade reliable nodes to maintain the segregation of duties. PoA is a PoW-based consensus, similar to the present hybrid consensus, that aims to produce a fork-free property and a lower variance of miners' payouts, thereby changing the basis of blockchain nonce that various puzzle solutions can be found each round [140]. For the first time, a blockchain-based consensus process supports different solutions called the 'generalized PoW'. Using a PBFT from the distributed system literature, all of these solutions are submitted to a committee without producing any fork.

Furthermore, they are all recorded, making the history of records difficult to falsify. The hybrid consensus concept and the generalized PoW build a fork-free hybrid consensus. It is worth noting that the hybrid protocol uses a blockchain to elect a miners' committee to validate transactions. On the other hand, the Fork-free hybrid consensus system allows the committee, rather than block proposers, to decide the record for the current round (containing transactions and accepted puzzle solutions) and future committee members once and for all.

In addition to PoW-based consensus algorithms, Algorand is a new cryptocurrency that promises to confirm transactions in under a minute. Algorand's core uses a Byzantine agreement mechanism called BA that scales to many users and allows Algorand to reach a consensus on a new block with low latency and no forks. Verifiable random functions (VRFs) to randomly choose users in a private and non-interactive manner are fundamental methods that render BA acceptable for Algorand. Weighted users, consensus by committee, cryptographic sorting, and participant replacement are some of the strategies used by Algorand to overcome these issues. The PoT consensus protocol, similar to Algorand, incorporates a trust component to fulfill the practical requirements of the service business, that is, to handle the untrustworthy behaviors that frequently occur in an open and public service network, in conjunction with incentive mechanisms. PoT is a hybrid blockchain solution that uses a consortium blockchain as the base deployment architecture. At the same time, the validation of the transactions of the consensus protocol occurs in an open and public network environment, demonstrating the fairness and impartiality characteristics of a public blockchain [118]. PoBT is a collection of innovative algorithms created for use with the IoT blockchain that validates transactions and blocks before they are recorded on the distributed ledger. For integrating the suggested consensus method with the Hyperledger Fabric architecture, PoBT is a fully functional solution. It is also a new kind of local service procedure to scale and find ways to deal with nodes that unexpectedly stall out.

The purpose of coins to preserve the blockchain is employed in the cryptocurrency community for so-called staking, i.e., PoS is one such method. Bentov et al. [138], [21] examine the pro-based blockchain architecture more formally, in conjunction with PoW as the sole mechanism for a blockchain protocol. Although Bentov et al. demonstrate that their protocols are secure against certain attacks, they do not present a formal model for analyzing PoS-based protocols or security proofs that rely on precise definitions. The concept of PoS is a natural alternative technique. Rather than investing computational resources in the leader election process, miners instead execute a process that selects one of them randomly, proportionally to the stake each holds according to the current blockchain ledger. By overcoming the Nothing-at-Stake barrier, PoS's feasibility is revealed. The endogenous value of the coin is required for the latter solution. A stakeholder in a blockchain is an agent who owns some of the network's currency. The Nothing-at-Stake problem implicitly presupposes that an agent's decision to update the blockchain does not affect the coin's value. This shows that this assumption is incorrect. Suppose that an agent adds to the blockchain in a way that promotes conflict. In that case, he incurs a penalty for all stakeholders since his action devalues blockchain coins as a medium of exchange to lower their value. Only stakeholders are authorized to change the PoS blockchain in PoS. Therefore, an agent incurs a cost if he updates the network in a way that causes continued contention [138]. In addition to the performance of the consensus mechanisms, we also describe the problem addressed in every study, as in Table 12. Therefore, with the analysis of 12 consensus mechanisms, we argue that PoS is a scalable consensus algorithm for the following reasons: 1) block generation rights are based on the stake rather than hash calculations with leading zeros, 2) block generation time is deficient, 3) energy efficient as no power is wasted in the number of hash calculations, and 4) low chance of fork gives stability to the blockchain. Based on the above four reasons, the PoS consensus mechanism can be considered to develop smart contracts for mobile devices.

RQ3: What are the shortcomings of the algorithms used in RQ1 and RQ2 that hinder the development of MSCs?

In this RQ, we aim to evaluate the scalability and consensus mechanism of the blockchain protocols identified in RQ1 and RQ2 as per Section V. We present the scalability approaches, the sharding protocols, and the scalable consensus algorithms in Tables 10, 11, and 12, respectively. ELASTICO is the first secure candidate for a sharding protocol for open blockchains that tolerates Byzantine opponents based on sharding. The transaction throughput of the blockchain is almost linearly proportional to the network's computing power when using ELASTICO. As the network increases, ELASTICO has virtually linear scalability in terms of computing capacity and does not require a quadratic number of messages. ELASTICO can handle adaptive Byzantine adversaries up to $f<n/4$, where f and n are the bounds of the adversarial and the total computational power, respectively. In addition, the TPS of the SSChain system increases linearly (from 26 to 6500) when the number of shards added to the network increases (0 to 50, respectively) without cross-shard transactions. When the number of shards exceeds 50, the network bandwidth is limited to 13 Mbps, and the TPS stops growing. Since root-chain miners must verify all shard blocks, the bandwidth use of root-chain miners and shard miners increases as the number of shards increases. In terms of scalability [110], the chainspace capacity increases linearly when additional shards are added, provided that transactions have a constant or sublinear number of inputs on average. Additionally, multiple nodes within the system must manage those inputs to ensure that the load of accepting transactions is divided between them. In continuation with the sharding approach, OmniLedger scale-out throughput with a 12.5% opponent, a shard size of 70, and several shards m ranging from 1 to 16. The number of shards has a nearly linear effect on the performance of ELASTICO [141].

Sharding, sharding with ledger pruning, and the committee-based approach perform transaction validation almost linear to the total nodes in the network. With literature analysis and reasoning, sharding and committee-based approaches are in our interest consideration because, in these approaches, the blockchain and P2P network divide into disjoint sets, each capable of processing only transactions associated with that set. The performance of the sharding protocols is better than that of other approaches, and ledger pruning sharding has more advantages because it addresses storage scalability. Therefore, sharding protocols could be the best option to develop MSCs while working on the following issues:

1) CONNECTION MANAGEMENT

All the sharding protocols have not included connection management from one shard to another in the mobile computing environment. The latter is because all algorithms are not designed for a mobile computing environment. In a mobile environment, each mobile device is always connected to a particular cell; a cell is a basic service area, and the cell of a mobile device changes with moving to a different location [142]. Based on the location of the cell, if a mobile device is in Shard 1, after some time, the device moves to another region, then the shard of that device must be changed. The latter could be possible with connection management in the mobile environment. Furthermore, the management of cross-sharing communication must be updated according to the mobile environment.

2) STRONG RESILIENCY

Resiliency is the resistance to prevent a fraction of malicious nodes in the P2P network. To represent resilience, let n be the total number of nodes and t be the number of malicious nodes on the P2P network. In the existing algorithm, Elastico, SSChain, and Zilliqa provide resiliency up to t<n/4, while RapidChain, OmniLedger, RepChain, and Rllerchain work with t<n/3 resiliency. The above algorithms apply t<n/4, t<n/3 on a small set of nodes in the whole network, e.g., 600 nodes in RapidChain. If the network grows with more nodes, the malicious nodes increase, which may harm the network. In the case of weak resiliency, such as tolerance 25% or 33% to Byzantine faults, the chances of network failure may increase in a short period of time [110]. Therefore, strong resiliency is required to prevent the shard and root chain from the effect of malicious nodes.

3) PoS AS CONSENSUS MECHANISM

Consensus is the core part of distributed computing to agree on the set of transactions, that is, a block. All sharding protocols (as in Table 11) use PoW and PBFT in the root and shard chains, respectively, except Algorand. The latter uses VRF for the leader election from the weighted users to add a block to the blockchain. PoW computation with a mobile device takes a long time, even in days. Additionally, PoW is not a scalable consensus mechanism.

Although the Bitcoin protocol has persuaded miners to maintain consistent copies of the transaction ledger, it is debatable whether this ledger can be called "decentralized." At the time of writing, four mining pools account for more than half of all blocks, while six mining pools account for more than three-quarters. Although a single firm does not own a mining pool's hardware, a recent study discovered that 11 "major mining groups" control more than half of the world's mining capacity. In principle, these miners have the ability to freeze any user's funds or delete earlier transactions from the ledger [143]. Therefore, the current state of the PoW and PBFT sharding protocols is not a suitable combination for the mobile computation of blockchains.

Furthermore, the PoBT uses the same strategy while proposing the block to trade as an incentive mechanism for the next block. Moreover, PoS is a scalable and energy-efficient consensus algorithm [144], as shown in Table 12. In this table, most studies use PoS with a combination of other algorithms such as PoW, PoT, DPoS, etc. The performance of the PoS algorithm is better than other existing algorithms (as in Table 12). However, all sharding protocols do not use the PoS consensus mechanism because oligopoly formation is the central issue. PoS is essentially a shareholder corporation in which the wealthy dominate since they control more assets and can manufacture new currencies more quickly than less fortunate participants. As a result, the (already) rich become even wealthier; this is an undemocratic strategy [22]. The PoS algorithm works based on the stake of a node, and then a group of nodes combine their stake and take control of the whole network, leading to centralization. Oligopoly formation occurs due to a lack of incentivization of the stakes of the nodes. Therefore, after developing a mechanism to prevent the formation of oligopoly, PoS can be used as a consensus mechanism in sharding protocols.

The PoS consensus mechanism has several unique benefits that make it suitable for smart contracts, especially in a mobile context. We have discussed earlier that PoS is much more energy efficient than PoW consensus algorithms, which is crucial given the energy limitations common to mobile devices [144], [133]. PoS is also more suitable for the processing capabilities of mobile devices because it requires less computing power and does not include computationally complex challenges such as PoW [23], [145]. From a security perspective, PoS reduces the likelihood of an attack 51% because it requires the attacker to control 51% of all the cryptocurrency on the network [133]. Furthermore, MSCs must be scalable to accommodate a high volume of transactions, and PoS has higher transaction throughput than PoW [94]. Finally, PoS encourages decentralization by removing the risk of mining power concentration often seen with PoW by allowing any network participant who holds the coin to become a validator [132]. The PoS consensus method is the best option for MSCs despite potential drawbacks and the danger of centralization brought about by stake concentration.

VI. DISCUSSION

This section discusses the technical challenges and recommendations in the subsections. In Section VI-A, we discuss the technical challenges of scalability approaches described in Table 10, Section VI-B discusses the challenges of consensus protocols described in Table 12, and Section VI-D discusses the technical recommendations for MSCs using various scalability approaches and consensus mechanisms.

A. CHALLENGES OF SHARDING PROTOCOLS

Sharding protocols heavily rely on the consensus mechanism used in the protocol; e.g., Elastico, RepChain, and Rapidchain use PoW to establish the identity of a node, forming a node committee and PBFT in the committee to agree on a set of transactions that is, a shard block [110], [115]. The sharding algorithms in Table 11 contribute to significant transaction throughput. However, these algorithms have some limitations, such as committee size in Elastico, epoch randomness, etc. This section discusses limitations, challenges, and future research direction based on the results in Table 11 and Table 12.

Most sharding-based protocols use PoW and PBFT to run the consensus mechanism in the P2P network. The major limitation to considering the sharding protocols for mobile devices is PoW because hash calculation with a leading number of zeros is very hard. Another limitation is PBFT, which works only on a smaller set of nodes. If the network grows, more parallel shards are formed, which increases the data migration overhead and the probability of network failure. Of the selected sharding algorithms, three of them use transaction sharding, which forces the network nodes to store a full copy of the blockchain while verifying a small set of transactions (e.g., shard) [110], [112], [113].

A blockchain network can handle more transactions at once using sharding technology. Below are some factors for which sharding-based methods are more suitable for mobile environments.

1) REDUCED STORAGE NEEDS

Sharding separates the blockchain into smaller chunks, each handled by a different network node. As a result, a mobile device connected to a sharded blockchain network only needs to retain the data related to its specific shard rather than the entire blockchain. Thus, much less storage space is required, which makes it better suited for mobile devices with low storage capacities.

2) LESS COMPUTATIONAL RESOURCES

With sharding, each node processes only a portion of all transactions. Due to this division of labor, mobile devices must process fewer transactions than desktop or server computers, which typically have more powerful hardware. This reduces the computational load.

3) LOWER ENERGY CONSUMPTION

Because each node in a sharded system processes fewer transactions, there is a corresponding decrease in the energy needed to process and validate transactions. This may be advantageous for mobile devices that use batteries because it can result in longer battery life due to less energy use.

4) INCREASED SPEED

Because transactions can be performed in parallel, sharding can drastically shorten the time it takes to validate a transaction. On mobile devices, which are frequently used for real-time applications, this might enhance the user experience.

Sharding enhances the blockchain's ability to scale, which helps to sustain performance as a mobile application's user base expands. This is essential for mobile applications because they often have a large user base.

Although the sharding algorithms claim significant scalability and transaction latency, these algorithms still have some drawbacks. For example, Elastico rebuilds all committees in every epoch and must solve PoW by every node to re-establish their identities. Aside from massive data migration and communication overhead, this process increases latency, as nodes need more time to solve PoW rather than validate the next transaction set. The committee size in Elastico is minimal (approximately 100 nodes) because PBFT yields significant performance in a small committee size that increases the failure probability of the protocol. Elastico, RollerChain, RepChain, and Zilliqa apply transaction sharding to verify the transaction, and each node has to maintain a full copy of the ledger, which increases the storage overhead. RapidChain also uses PoW to join the network, the gossiping protocol, and synchronous consensus to broadcast the message and consensus. The gossiping protocol requires more time to broadcast the message, and synchronous consensus takes more time, thus increasing the broadcast latency. The cross-shard transaction increases in RapidChain due to the partitioning of transactions based on the transaction ID. OmniLedger can tolerate only t<n/4 byzantine adversary, the same as in Elastico. This protocol has the same broadcast overhead as in RapidChain for each block of transactions. A trusted setup is required to generate an initial configuration to send the VRF in the first epoch. In all shard-based algorithms, Algorand does not use PoW; instead, it uses VRF to select weighted members to form a committee randomly.

5) CROSS-SHARD COMMUNICATION OVERHEAD

Sharding is a promising solution to enhance the scalability of blockchain networks. However, it also brings its own set of challenges and limitations, particularly with regard to cross-shard communication. The primary challenge is to maintain consistency and atomicity across shards during transactions. Coordinating the execution of transactions that involve multiple shards and maintaining a consistent state across the shards is a complex task. This complexity increases latency and potential throughput bottlenecks, as cross-shard communication requires synchronization and verification processes. Passing messages between shards also poses a significant challenge, requiring efficient mechanisms to manage the communication overhead and preserve the scalability benefits of sharding. Cryptographic proofs and inter-shard validation mechanisms become imperative to prevent potential exploits during cross-shard transactions, leading to security and integrity concerns. Load balancing among shards is also an issue, as uneven transaction distribution may lead to congestion in specific shards, offsetting the intended scalability gains. Ensuring data availability for cross-shard transactions, especially when shards need to access data stored in others, is a non-trivial task. Fault tolerance in cross-shard communication adds complexity and requires robust mechanisms to address potential shard failures.

Moreover, coordinating the execution of smart contracts across shards introduces complexities in managing the contract state and execution. Designing protocols for cross-shard communication is intricate, requiring careful consideration of various challenges and trade-offs. It is critical to address these challenges to realize the full potential of sharding to achieve scalable and efficient blockchain networks. Ongoing research and development efforts in the blockchain community are focused on finding innovative solutions to these challenges and refining shard-based systems.

B. CHALLENGES OF SCALABLE CONSENSUS PROTOCOLS

The performance of the PoS consensus is better compared to other algorithms (see Table 12). PoS is also an energy-efficient and scalable algorithm compared to PoW [128]. In the study SS13 [23], the author proposes a blockchain model that uses the PoS consensus mechanism to reduce the chances of the fork, achieving consensus without wasting energy resources, and finally, a probabilistic model to handle problems that do not lie. Bentov et al. [21] proposed the PoS consensus protocol prior to the above study, introducing pure PoS into the PPCoin system. PPCoin system uses PoW to supply the initial coin supply and then PoS to create the next block based on the unspent coin and time weight of these coins. The authors also discuss the pure PoS algorithm to solve the rational fork problem, which takes approximately four days to solve the forks on the blockchains. Another category is hybrid consensus, where the components of two more algorithms are mixed to produce a hybrid algorithm with more robust features and significant performance. These algorithms are: Hybrid Mining [128], Ethereum's Hybrid Casper Protocol [132], TwinsCoin [134], DPoS with Downgrade [100], PoBT [26], and flexible PoA [146]. These hybrid algorithms require PoW to initiate the consensus process that limits their application. Additionally, these algorithms increase the overhead of maintaining two types of blocks, one for PoW and the other for PoS. The DPoS with the downgrade mechanism limits the effect of a malicious node to ensure security and normal operation on the network. In selected studies, eight consensus mechanisms use the PoS algorithm for block generation, but PoS has a low adoption rate in the blockchain industry. Currently, two major systems in the blockchain industry (Bitcoin and Ethereum) still use PoW because PoW is more secure than PoS. After addressing issues in the PoS, such as the nothing-at-stake problem and oligopoly formation, this consensus can be used in the blockchain industry because the PoS consensus algorithm is scalable, energy efficient, and has faster block generation capability with low latency.

Recently, the study [147] has suggested the architecture of a universal tokenization platform called Alphabill, which allows tokenization, transfer, and exchange of universal assets as a global medium of exchange. Alphabill has been designed genuinely to tokenize universal assets [147]. As such, it shares objectives with federated blockchain technologies such as Polkadot,[5] which is designed as a ''heterogeneous multichain'' [148]. The key difference between Alphabill and Polkadot is in their approach to decomposition. Polkadot is a federation of multiple blockchains, whereas Alphabill is a single-partitioned blockchain. The objective of universal asset tokenization can be detected in the design decisions and innovations of the Alphabill platform [147], that is, 1) systematic support for joining a transaction system to the platform, 2) systematic features for the interaction of hosted tokens, 3) uncapped scalability.

Sharding introduces complexity in distributed protocols and algorithms to handle tasks like transaction routing, membership management, and cross-shard commits. This increases the development effort and makes formal verification more challenging. Most consensus protocols scale poorly with the number of validators, which limits the scalability of sharding. PBFT requires quadratic communication, increasing consensus overhead rapidly beyond hundreds of nodes. PoW scales better but is limited by block intervals. In summary, sharding and consensus protocols face several key challenges, including cross-shard coordination, distributed protocol complexity, scalability bottlenecks in consensus, takeover attacks on shards, stake centralization in PoS, network partitioning issues, and small singleton shards. Ongoing research is being conducted to address these limitations.

C. SECURITY ANALYSIS OF SCALABLE CONSENSUS PROTOCOLS

When considering scalable consensus algorithms, it is essential to consider security concerns. We considered 51% attack, Stake Centralization, and Nothing at stake for analysis since security analysis is carried out in selected scalable consensus algorithms (refer V-C). PoW algorithm is vulnerable to the 51% attack because if the miner collects 51% hash power on the network, then the miner has significant block generation rights. In PoS, miners have a 51% stake attack in place of hash power, which happened in Pure PoS. However, this attack is mitigated when using PoS with mechanisms such as randomized validation selection and periodic reselection of validators. A stake centralization attack occurs in PoS when a few nodes influence the networks by collecting significant stakes of the network. We have already discussed this in Section V-C. This attack is mitigated by protocols using hybrid PoW/PoS consensus or some mechanism, such as randomized committee selection in Algorand and periodic validator selection in RapidChain. Nothing at stake is another attack on PoS consensus where nodes can validate multiple chains without penalty. The attack is minimized by penalizing the malicious nodes in the network.

D. TECHNICAL RECOMMENDATIONS FOR MSCS

Table 13 summarizes different technological recommendations for improving MSC performance and utility.

[5] https://polkadot.network/

TABLE 13. Technical recommendations for implementation of mobile smart-contracts to enhance scalability and oligopoly minimization.

Recommendation ID	Scalability Approach / Consensus Mechanism	Technical Actions	Oligopoly Mitigation	Mobile Smart Contract Suitability
1	Sharding	Reduce the workload on each node by distributing data across several nodes by state sharding	Low impact	High
2	Ledger Pruning	Ledger pruning eliminate archival transactions and states	Low impact	Moderate
3	Committee-based	Algorand's committee-based strategy increases transaction throughput	Committee rotation to prevent centralization	High
4	Plasma	Create separate chains for particular smart contracts and combine the results into the main chain	Low impact	Moderate
5	State Channels	Reduce the on-chain load by using state channels for off-chain computations and updates	Low impact	High
6	PoS with Dynamic Validator Selection	Implement PoS, however, permit more frequent rotation of validators based on criteria other than stake	High impact	High
7	Hybrid Consensus	Combine both PoW and PoS to create blocks	Moderate impact	Moderate
8	Proof-of-Trust	Consensus votes, user reputation, and trust scores are key parameters for consensus mechanism	Moderate impact	High
9	Ouroboros	Randomized leader selection feature of PoS-based Ouroboros to reduce the impact of grinding attacks	High impact	Moderate
10	DDPoS with Downgrade	DDPoS implementation to reduce the priority of malicious nodes	High impact	Low

Sharding improves throughput by parallelizing network activity, making it ideal for MSCs. It reduces the demands on individual nodes for data storage and processing, which is excellent for mobile devices with limited resources. However, because powerful entities may control numerous shards, sharding may not immediately decrease the risk of oligopoly.

The *Ledger Pruning method* removes old and useless transactions from the blockchain ledger, an essential feature for MSCs given the often low storage capabilities of mobile devices. While this solution helps promote the mobility of the blockchain for mobile nodes, it has no impact on the oligopoly problem.

Plasma improves scalability by making it easier to create secondary blockchains linked to the parent blockchain. Because of their independence from the main chain, these sub-chains can execute transactions significantly more swiftly and cheaply. However, Plasma does not directly address its off-chain processing support.

State channels allow off-chain transactions, with only the final state committed to the main blockchain. This method significantly speeds up transaction processing and is especially helpful for MSCs because it reduces the computing stress on mobile devices.

In the *committee-based approach*, each block is validated by a small, randomly selected set of nodes. This approach dramatically increases throughput, making it ideal for MSCs. Additionally, rotating committee members at random and regularly helps reduce the risk of oligopoly by preventing a single firm from monopolizing the network. Traditional *PoS* systems can exacerbate oligopolistic tendencies by giving more power to nodes with more significant stakes. However, adding a dynamic validation system selection and considering factors other than stake can solve this issue. This open technique allows nodes with lower stakes to participate in the validation process, especially for MSCs, where lightweight nodes can serve as validators.

Combining PoW with PoS to develop a hybrid consensus method can result in a more balanced system. This hybrid strategy strikes a balance between decentralization and scalability. Although its direct impact on oligopoly reduction may be limited, it presents a level of appropriateness for MSCs worth exploring.

PoT is a consensus process incorporating reputation or trust measurements, skewing the influence toward nodes with higher trust ratings. PoT is appropriate for MSCs because of its lower processing demands and the possibility of reducing oligopoly using various decision criteria.

Ouroboros, a PoS-based consensus method that uses randomized leader selection, reduces the possibility of a concentrated number of validators dominating the network.

Although its relevance to MSCs is limited, its effectiveness in reducing oligopoly is significant.

Finally, *DDPoS* includes a mechanism to remove or eliminate malicious or underperforming nodes. This strengthens the resilience and adaptability of the system in the face of oligopoly and centralization. DDPoS, on the other hand, may not be the best solution for MSC implementations due to its complexity.

E. SHARDING AND SCALABLE CONSENSUS PROTOCOLS FOR MSCs

Sharding and scalable consensus algorithms are beneficial for MSCs because they are essential to address the scalability challenges in blockchain for mobile environments. The splitting partitions the blockchain regarding size, transactions, and computing capabilities so that each shard can process independently. Parallel processing of each shard significantly increases throughput and latency. Scalable consensus algorithms, such as variants of PoS, hybrid PoW, and PoS, etc., have scalability advantages over traditional PoW. These algorithms are designed to achieve consensus more efficiently to improve transaction speed and overall performance while maintaining system security. We have already discussed sharding techniques in Section V-B and consensus algorithms in Section V-C in detail.

VII. CONCLUSION

The objective of this paper is to analyze different approaches to scalability and consensus mechanisms described in the literature and to identify a scalable approach with the PoS algorithm to develop MSCs.This paper analyzes different approaches to scalability and consensus mechanisms described in the literature. We present an SLR on scalable consensus mechanisms and scalability issues for developing MSCs. In this paper, a total of 2073 papers are identified for scalability and consensus mechanisms. Of these, 25 papers were selected through an exhaustive process. By critically analyzing the selected studies, we discover that the sharding protocols need to include connection management and strong resiliency. Also, the PoS is a scalable consensus mechanism; however, it has issues of the oligopoly formation and nothing-at-stake problem. The major limitation in the sharding and consensus algorithms is resource incentives. We have discussed that the PoS consensus mechanism is a significant option for MSCs with sufficient evidence. The other limitations in sharding algorithms are the small shard or committee size and inefficient cross-shard communication. Contributions of this SLR are (1) a classification of scalability approaches based on throughput and consensus mechanism used, (2) an identification and characterization of sharding protocols based on various parameters such as sharding type, consensus mechanism, number of nodes, resiliency, latency, throughput, and cross-shard communication, and (3) an identification of scalable consensus algorithms on various performance parameters. We argue that a deep understanding of scalability and consensus algorithms is critical in improving the scalability of blockchain-based smart contract solutions. Furthermore, developing a blockchain-based smart contract solution for a mobile device is essential to gain significant scalability. With efficient consensus algorithms, the latter promises to reduce the operating cost of a blockchain network. We discuss the real-world impacts of smart contracts in a mobile environment. In future work, we would like to develop a mobile smart contract by applying the sharding protocols while employing PoS as a consensus mechanism to validate the transactions. This study provides the future direction for 1) researchers with a preference for where and how to start their literature studies in the area of scalability of blockchains and smart contracts, 2) developers who aim to find solutions to scaling blockchains, and 3) users, who want to be more aware of the vulnerabilities in consensus mechanisms and existing tools.

REFERENCES

[1] J. Wiles (Feb. 15, 2022). *What is Web3*. [Online]. Available: https://www.gartner.com/en/articles/what-is-web3

[2] T. Stackpole. (May 2022). *What is Web3*. Harvard Bus. Review. [Online]. Available: https://hbr.org/2022/05/what-is-web3

[3] L. Jin and K. Parrott, "Web3 is our chance to make a better internet," *Harvard Bus. Rev.*, vol. 10, pp. 1–12, May 2022. [Online]. Available: https://hbr.org/2022/05/web3-is-our-chance-to-make-a-better-internet

[4] J. Esber and S. D. Kominers, "Why build in Web3," *Harvard Bus. Rev.*, vol. 16, pp. 1–38, May 2022. [Online]. Available: https://hbr.org/2022/05/why-build-in-web3

[5] G. Edelman, "Paradise at the crypto arcade," *Wired*, pp. 49–59, Jun. 2022. Accessed: Feb. 27, 2024. [Online]. Available: https://www.wired.com/story/web3-paradise-crypto-arcade/

[6] A. Buldas, D. Draheim, M. Gault, and M. Saarepera, "Towards a foundation of Web3," in *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications* (Communications in Computer and Information Science), vol. 1688, T. K. Dang, J. Küng, and T. M. Chung, Eds. Singapore: Springer, Nov. 2022, pp. 3–18.

[7] *Measuring Digital Development: Facts and Figures 2022*, International Telecommunications Union—Development Sector, ITU Publications, Geneva, Switzerland, 2022.

[8] NDTV. *UPI Hits Record 9 Billion Transactions*. Accessed: Feb. 27, 2024. [Online]. Available: https://www.ndtv.com/india-news/upi-hits-record-9-billion-transactions-worth-rs-14-lakh-crore-in-may-4086825

[9] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Decentralized Bus. Rev. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] V. Dwivedi, V. Pattanaik, V. Deval, A. Dixit, A. Norta, and D. Draheim, "Legally enforceable smart-contract languages: A systematic literature review," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–34, Jun. 2021.

[11] J. Huang, L. Kong, J. Wang, G. Chen, J. Gao, G. Huang, and M. K. Khan, "Secure data sharing over vehicular networks based on multi-sharding blockchain," *ACM Trans. Sensor Netw.*, vol. 20, no. 2, pp. 1–23, Mar. 2024.

[12] C. Shen and F. Pena-Mora, "Blockchain for cities—A systematic literature review," *IEEE Access*, vol. 6, pp. 76787–76819, 2018.

[13] A. Shahaab, B. Lidgey, C. Hewage, and I. Khan, "Applicability and appropriateness of distributed ledgers consensus protocols in public and private sectors: A systematic review," *IEEE Access*, vol. 7, pp. 43622–43636, 2019.

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

[15] Q. Lu, X. Xu, Y. Liu, I. Weber, L. Zhu, and W. Zhang, "UBaaS: A unified blockchain as a service platform," *Future Gener. Comput. Syst.*, vol. 101, pp. 564–575, Dec. 2019.

[16] Y. Lu, "The blockchain: State-of-the-art and research challenges," *J. Ind. Inf. Integr.*, vol. 15, pp. 80–90, Sep. 2019.

[17] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, Jun. 2020.

[18] T. Wu, G. Jourjon, K. Thilakarathna, and P. L. Yeoh, "MapChain-D: A distributed blockchain for IIoT data storage and communications," *IEEE Trans. Ind. Informat.*, vol. 19, no. 9, pp. 9766–9776, Jan. 2023.

[19] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-based federated learning for securing Internet of Things: A comprehensive survey," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–43, Sep. 2023.

[20] M. Khalid, S. Hameed, A. Qadir, S. A. Shah, and D. Draheim, "Towards SDN-based smart contract solution for IoT access control," *Comput. Commun.*, vol. 198, pp. 1–31, Jan. 2023.

[21] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Financial Cryptography Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Germany: Springer, Aug. 2016, pp. 142–157.

[22] M. Borge, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, and B. Ford, "Proof-of-personhood: Redemocratizing permissionless cryptocurrencies," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Apr. 2017, pp. 23–26.

[23] F. Saleh, "Blockchain without waste: Proof-of-stake," *Rev. Financial Stud.*, vol. 34, no. 3, pp. 1156–1190, Feb. 2021.

[24] L. Chen, L. Xu, Z. Gao, Y. Lu, and W. Shi, "Protecting early stage Proof-of-Work based public blockchain," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2018, pp. 122–127.

[25] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Gener. Comput. Syst.*, vol. 107, pp. 760–769, Jun. 2020.

[26] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "PoBT: A lightweight consensus algorithm for scalable IoT business blockchain," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2343–2355, Mar. 2020.

[27] S. Siddiqui, S. Hameed, S. A. Shah, A. K. Khan, and A. Aneiba, "Smart contract-based security architecture for collaborative services in municipal smart cities," *J. Syst. Archit.*, vol. 135, Feb. 2023, Art. no. 102802.

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

[29] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[30] K. Croman, "On scaling decentralized blockchains," in *Proc. Int. Conf. Financ. Cryptogr. Data Security*, 2016, pp. 106–125.

[31] W. Dai, D. Xiao, H. Jin, and X. Xie, "A concurrent optimization consensus system based on blockchain," in *Proc. 26th Int. Conf. Telecommun. (ICT)*, Apr. 2019, pp. 244–248.

[32] A. Arooj, M. S. Farooq, and T. Umer, "Unfolding the blockchain era: Timeline, evolution, types and real-world applications," *J. Netw. Comput. Appl.*, vol. 207, Nov. 2022, Art. no. 103511.

[33] A. I. Sanka and R. C. C. Cheung, "A systematic review of blockchain scalability: Issues, solutions, analysis and future research," *J. Netw. Comput. Appl.*, vol. 195, Dec. 2021, Art. no. 103232.

[34] H. Chen and Y. Wang, "MiniChain: A lightweight protocol to combat the UTXO growth in public blockchain," *J. Parallel Distrib. Comput.*, vol. 143, pp. 67–76, Sep. 2020.

[35] A. I. Sanka and R. C. C. Cheung, "Efficient high performance FPGA based NoSQL caching system for blockchain scalability and throughput improvement," in *Proc. 26th Int. Conf. Syst. Eng. (ICSEng)*, Dec. 2018, pp. 1–8, doi: 10.1109/ICSENG.2018.8638204.

[36] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Comput. Surv.*, vol. 52, no. 3, Jul. 2019, Art. no. 51.

[37] X. Li, Y. Wang, Y. Ding, S. Ma, B. Xiao, Z. Guo, X. Kang, X. Ma, and J. Mai, "A privacy-preserving lightweight energy data sharing scheme based on blockchain for smart grid," in *Proc. 18th Int. Conf. Collaborative Comput., Netw., Appl. Worksharing (CollaborateCom)*, vol. 461, H. Gao, X. Wang, W. Wei, and T. Dagiuklas, Eds. Cham, Switzerland: Springer, Jan. 2022, pp. 91–110.

[38] W. Viriyasitavat, L. D. Xu, Z. Bi, D. Hoonsopon, and N. Charoenruk, "Managing QoS of Internet-of-Things services using blockchain," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1357–1368, Dec. 2019.

[39] M. de Vos, G. Ishmaev, and J. Pouwelse, "Decentralizing components of electronic markets to prevent gatekeeping and manipulation," *Electron. Commerce Res. Appl.*, vol. 56, Nov. 2022, Art. no. 101220.

[40] Z. Guo, Z. Gao, Q. Liu, C. Chakraborty, Q. Hua, K. Yu, and S. Wan, "RNS-based adaptive compression scheme for the block data in the blockchain for IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 9239–9249, Dec. 2022.

[41] A. R. Sai, J. Buckley, B. Fitzgerald, and A. L. Gear, "Taxonomy of centralization in public blockchain systems: A systematic literature review," *Inf. Process. Manage.*, vol. 58, no. 4, Jul. 2021, Art. no. 102584.

[42] K. Y. Yap, H. H. Chin, and J. J. Klemeš, "Blockchain technology for distributed generation: A review of current development, challenges and future prospect," *Renew. Sustain. Energy Rev.*, vol. 175, Apr. 2023, Art. no. 113170.

[43] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 243–252.

[44] S. Aggarwal and N. Kumar, "Attacks on blockchain," *Adv. Comput.*, vol. 121, pp. 399–410, Jan. 2021.

[45] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *Financial Cryptography and Data Security*, R. Böhme and T. Okamoto, Eds. Berlin, Heidelberg: Springer, Jan. 2015, pp. 528–547.

[46] A. Mohsenzadeh, A. Jalaly Bidgoly, and Y. Farjami, "A fair consensus model in blockchain based on computational reputation," *Expert Syst. Appl.*, vol. 204, Oct. 2022, Art. no. 117578.

[47] A. Mohsenzadeh, A. J. Bidgoly, and Y. Farjami, "A novel reputation-based consensus framework (RCF) in distributed ledger technology," *Comput. Commun.*, vol. 190, pp. 126–144, Jun. 2022.

[48] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2017, pp. 464–467.

[49] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[50] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," in *Proc. Int. Conf. Princ. Secur. Trust*. Cham, Switzerland: Springer, 2017, pp. 164–186.

[51] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd USENIX Symp. Operating Syst. Design Implement.*, 1999, pp. 173–186.

[52] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: A tutorial," *ACM Comput. Surveys*, vol. 22, no. 4, pp. 299–319, Dec. 1990.

[53] D. Gkorou, J. Pouwelse, and D. Epema, "Trust-based collection of information in distributed reputation networks," in *Proc. 30th Annu. ACM Symp. Appl. Comput.* New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 2312–2319.

[54] S. Kudva, S. Badsha, S. Sengupta, I. Khalil, and A. Zomaya, "Towards secure and practical consensus for blockchain based VANET," *Inf. Sci.*, vol. 545, pp. 170–187, Feb. 2021.

[55] R. Delaviz, J. Pouwelse, and D. Epema, "Targeted and scalable information dissemination in a distributed reputation mechanism," in *Proc. 7th ACM Workshop Scalable Trusted Comput.* New York, NY, USA: Association for Computing Machinery, Oct. 2012, pp. 55–66.

[56] D. Bradbury, "In blocks [security Bitcoin]," *Eng. Technol.*, vol. 10, no. 2, pp. 68–71, Mar. 2015.

[57] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, "Demystifying incentives in the consensus computer," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 706–719.

[58] A. K. Yadav, K. Singh, A. H. Amin, L. Almutairi, T. R. Alsenani, and A. Ahmadian, "A comparative study on consensus mechanism with security threats and future scopes: Blockchain," *Comput. Commun.*, vol. 201, pp. 102–115, Mar. 2023.

[59] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin," in *Proc. 19th Int. Conf. Financial Cryptogr. Data Secur.*, 2015, pp. 507–527.

[60] P. Dai, N. Mahi, J. Earls, and A. Norta. (2017). *Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform*. Qtum Found. [Online]. Available: https://www.cryptoground.com/storage/files/1527488954_a2772efe4dc8ed1100319c6480195fb1.pdf

[61] V. Buterin, "A next-generation smart contract and decentralized application platform," *Ethereum*, vol. 3, no. 37, pp. 1–36, 2014.

[62] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, pp. 1–9, Sep. 1997.

[63] G. Laatikainen, M. Li, and P. Abrahamsson, "A system-based view of blockchain governance," *Inf. Softw. Technol.*, vol. 157, May 2023, Art. no. 107149.

[64] K. Bhargavan, "Formal verification of smart contracts: Short paper," in *Proc. ACM Workshop Program. Lang. Anal. Secur.*, 2016, pp. 91–96.

[65] C. Sillaber and B. Waltl, "Life cycle of smart contracts in blockchain ecosystems," *Datenschutz und Datensicherheit DuD*, vol. 41, no. 8, pp. 497–500, Aug. 2017.

[66] A. Norta, "Designing a smart-contract application layer for transacting decentralized autonomous organizations," in *1st Int. Conf. Adv. Comput. Data Sci. (ICACDS)*, vol. 461, M. Singh, P. Gupta, V. Tyagi, A. Sharma, T. Ören, and W. Grosky, Eds. Singapore: Springer, Jul. 2017, pp. 595–604.

[67] V. Dwivedi, V. Deval, A. Dixit, and A. Norta, "Formal-verification of smart-contract languages: A survey," in *Proc. 3rd Int. Conf. Adv. Comput. Data Sci. (ICACDS)*, vol. 1046, P. Gupta, V. Tyagi, J. Flusser, T. Ören, and R. Kashyap, Eds. Singapore: Springer, Jul. 2019, pp. 738–747.

[68] A. Dixit, V. Deval, V. Dwivedi, A. Norta, and D. Draheim, "Towards user-centered and legally relevant smart-contract development: A systematic literature review," *J. Ind. Inf. Integr.*, vol. 26, Mar. 2022, Art. no. 100314.

[69] V. K. Dwivedi and A. Norta, "A legally relevant socio-technical language development for smart contracts," in *Proc. IEEE 3rd Int. Workshops Found. Appl. Self* Syst. (FAS*W)*, Sep. 2018, pp. 11–13.

[70] H. Zhang, J. Wang, and Y. Ding, "Blockchain-based decentralized and secure keyless signature scheme for smart grid," *Energy*, vol. 180, pp. 955–967, Aug. 2019.

[71] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009.

[72] B. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," Keele Univ., Durham, Keele, U.K., Tech. Rep. EBSE-2007-01, Jul. 2007. [Online]. Available: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

[73] M. Bartoletti and L. Pompianu, "An empirical analysis of smart contracts: Platforms, applications, and design patterns," in *Proc. 21st Int. Conf. Financial Cryptography Data Secur. (FC)*, vol. 10323, M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. Ryan, V. Teague, A. Bracciali, M. Sala, F. Pintore, and M. Jakobsson, Eds. Cham, Switzerland: Springer, Nov. 2017, pp. 494–509.

[74] E. F. Jesus, V. R. L. Chicarino, C. V. N. de Albuquerque, and A. A. D. A. Rocha, "A survey of how to use blockchain to secure Internet of Things and the stalker attack," *Secur. Commun. Netw.*, vol. 2018, pp. 1–27, Apr. 2018.

[75] Q. Bao, B. Li, T. Hu, and X. Sun, "A survey of blockchain consensus safety and security: State-of-the-art, challenges, and future work," *J. Syst. Softw.*, vol. 196, Feb. 2023, Art. no. 111555.

[76] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor, "Blockchain and scalability," in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2018, pp. 122–128.

[77] M. Bez, G. Fornari, and T. Vardanega, "The scalability challenge of Ethereum: An initial quantitative analysis," in *Proc. IEEE Int. Conf. Service-Oriented Syst. Eng. (SOSE)*, Apr. 2019, pp. 167–176.

[78] J. Göbel and A. E. Krzesinski, "Increased block size and Bitcoin blockchain dynamics," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–6, doi: 10.1109/ATNAC.2017.8215367.

[79] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 1545–1550.

[80] D. K. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. Njilla, "Consensus protocols for blockchain-based data provenance: Challenges and opportunities," in *Proc. IEEE 8th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2017, pp. 469–474.

[81] S. S. Hazari and Q. H. Mahmoud, "Comparative evaluation of consensus mechanisms in cryptocurrencies," *Internet Technol. Lett.*, vol. 2, no. 3, p. e100, May 2019.

[82] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Consensus in the age of blockchains," 2017, *arXiv:1711.03936*.

[83] W. Wang, D. Niyato, P. Wang, and A. Leshem, "Decentralized caching for content delivery based on blockchain: A game theoretic perspective," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422547.

[84] A. Kalla, C. de Alwis, P. Porambage, G. Gür, and M. Liyanage, "A survey on the use of blockchain for future 6G: Technical aspects, use cases, challenges and research directions," *J. Ind. Inf. Integr.*, vol. 30, Nov. 2022, Art. no. 100404.

[85] M. H. Nasir, J. Arshad, M. M. Khan, M. Fatima, K. Salah, and R. Jayaraman, "Scalable blockchains—A systematic review," *Future Gener. Comput. Syst.*, vol. 126, pp. 136–162, Jan. 2022.

[86] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, and R. P. Liu, "Survey: Sharding in blockchains," *IEEE Access*, vol. 8, pp. 14155–14181, 2020.

[87] V. Pawar and S. Sachdeva, "CovidBChain: Framework for access-control, authentication, and integrity of COVID-19 data," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 28, Oct. 2022, Art. no. e7397.

[88] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, Sep. 2019.

[89] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125244–125262, 2020.

[90] D. Khan, L. T. Jung, and M. A. Hashmani, "Systematic literature review of challenges in blockchain scalability," *Appl. Sci.*, vol. 11, no. 20, p. 9372, Oct. 2021.

[91] B. Lashkari and P. Musilek, "A comprehensive review of blockchain consensus mechanisms," *IEEE Access*, vol. 9, pp. 43620–43652, 2021.

[92] S. Bouraga, "A taxonomy of blockchain consensus protocols: A survey and classification framework," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114384.

[93] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE Access*, vol. 7, pp. 28712–28725, 2019.

[94] V.-C. Nguyen, H.-L. Pham, T.-H. Tran, H.-T. Huynh, and Y. Nakashima, "Digitizing invoice and managing VAT payment using blockchain smart contract," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 74–77.

[95] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2567–2572.

[96] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jun. 2017, pp. 557–564.

[97] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018.

[98] B. A. Tama, B. J. Kweka, Y. Park, and K.-H. Rhee, "A critical review of blockchain and its current applications," in *Proc. Int. Conf. Electr. Eng. Comput. Sci. (ICECOS)*, Aug. 2017, pp. 109–113.

[99] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An overview of smart contract: Architecture, applications, and future trends," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 108–113.

[100] W. Yang, E. Aghasian, S. Garg, D. Herbert, L. Disiuta, and B. Kang, "A survey on blockchain-based Internet service architecture: Requirements, challenges, trends, and future," *IEEE Access*, vol. 7, pp. 75845–75872, 2019.

[101] D. Yang, C. Long, H. Xu, and S. Peng, "A review on scalability of blockchain," in *Proc. 2nd Int. Conf. Blockchain Technol.* New York, NY, USA: Association for Computing Machinery, May 2020, pp. 1–6, doi: 10.1145/3390566.3391665.

[102] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[103] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.

[104] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.

[105] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *Proc. 4th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Jan. 2017, pp. 1–5, doi: 10.1109/ICACCS.2017.8014672.

[106] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1432–1465, 2nd Quart., 2020.

[107] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[108] P. Otte, M. de Vos, and J. Pouwelse, "TrustChain: A Sybil-resistant scalable blockchain," *Future Gener. Comput. Syst.*, vol. 107, pp. 770–780, Jun. 2020.

[109] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, and P. Hochschild, "Spanner: Googles globally distributed database," *ACM Trans. Comput. Syst. (TOCS)*, vol. 31, no. 3, Aug. 2013, Art. no. 8.

[110] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, USA: ACM Press, Oct. 2016, pp. 17–30.

[111] QuarkChain Team. (2017). *QuarkChain: A High-Capacity Peer-to-Peer Transactional System*. [Online]. Available: https://quarkchain.io/

[112] Z. Team. (2017) *The Zilliqa Technical Whitepaper*. [Online]. Available: https://docs.zilliqa.com/whitepaper.pdf

[113] X. Feng, J. Ma, Y. Miao, Q. Meng, X. Liu, Q. Jiang, and H. Li, "Pruneable sharding-based blockchain protocol," *Peer Peer Netw. Appl.*, vol. 12, no. 4, pp. 934–950, Jul. 2019.

[114] C. Huang, Z. Wang, H. Chen, Q. Hu, Q. Zhang, W. Wang, and X. Guan, "RepChain: A reputation-based secure, fast, and high incentive blockchain system via sharding," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4291–4304, Mar. 2021.

[115] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 931–948.

[116] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598.

[117] H. Chen and Y. Wang, "SSChain: A full sharding protocol for public blockchain without data migration overhead," *Pervas. Mobile Comput.*, vol. 59, Oct. 2019, Art. no. 101055.

[118] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine agreements for cryptocurrencies," in *Proc. 26th Symp. Operating Syst. Princ.*, Oct. 2017, pp. 51–68.

[119] J. Poon and V. Buterin. (2017). *Plasma: Scalable Autonomous Smart Contracts*. [Online]. Available: https://plasma.io/plasma-deprecated.pdf

[120] S. Dziembowski, L. Eckey, S. Faust, J. Hesse, and K. Hostáková, "Multi-party virtual state channels," in *Advances in Cryptology EUROCRYPT 2019*, vol. 11476. Cham, Switzerland: Springer, Apr. 2019, pp. 625–656.

[121] S. Dziembowski, S. Faust, and K. Hostáková, "General state channel networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 949–966.

[122] C. Molina-Jimenez, E. Solaiman, I. Sfyrakis, I. Ng, and J. Crowcroft, "On and off-blockchain enforcement of smart contracts," in *Proc. Euro-Par Parallel Process. Workshops*. Cham, Switzerland: Springer, Dec. 2019, pp. 342–354.

[123] M. Memon, S. S. Hussain, U. A. Bajwa, and A. Ikhlas, "Blockchain beyond Bitcoin: Blockchain technology challenges and real-world applications," in *Proc. Int. Conf. Comput., Electron. Commun. Eng. (iCCECE)*, Aug. 2018, pp. 29–34.

[124] S. Kalra, R. Sanghi, and M. Dhawan, "Blockchain-based real-time cheat prevention and robustness for multi-player online games," in *Proc. 14th Int. Conf. Emerg. Netw. Exp. Technol.* New York, USA: Association for Computing Machinery, Dec. 2018, pp. 178–190.

[125] R. M. A. Latif, S. Iqbal, O. Rizwan, S. U. A. Shah, M. Farhan, and F. Ijaz, "Blockchain transforms the retail level by using a supply chain rules and regulation," in *Proc. 2nd Int. Conf. Commun., Comput. Digit. Syst. (C-CODE)*, Mar. 2019, pp. 264–269.

[126] A. Norta, A. Kormiltsyn, C. Udokwu, V. Dwivedi, S. Aroh, and I. Nikolajev, "A blockchain implementation for configurable multi-factor challenge-set self-sovereign identity authentication," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Aug. 2022, pp. 455–461.

[127] Y. Zhu, G. Zheng, and K.-K. Wong, "Blockchain-empowered decentralized storage in air-to-ground industrial networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3593–3601, Jun. 2019.

[128] K. Chatterjee, A. K. Goharshady, and A. Pourdamghani, "Hybrid mining: Exploiting blockchain's computational power for distributed problem solving," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.* New York, NY, USA: ACM Press, Apr. 2019, pp. 374–381.

[129] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Trans. Services Comput.*, vol. 12, no. 3, pp. 429–445, May 2019.

[130] J. Chen and S. Micali, "Algorand: A secure and efficient distributed ledger," *Theor. Comput. Sci.*, vol. 777, pp. 155–183, Jul. 2019.

[131] S. Cho, S. Y. Park, and S. R. Lee, "Blockchain consensus rule based dynamic blind voting for non-dependency transaction," *Int. J. Grid Distrib. Comput.*, vol. 10, no. 12, pp. 93–106, Dec. 2017.

[132] V. Buterin, D. Reijsbergen, S. Leonardos, and G. Piliouras, "Incentives in Ethereum's hybrid Casper protocol," *Int. J. Netw. Manage.*, vol. 30, no. 5, Feb. 2020, Art. no. e2098.

[133] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Proc. Annu. Int. Cryptol. Conf.*, Jul. 2017, pp. 357–388.

[134] T. Duong, A. Chepurnoy, L. Fan, and H.-S. Zhou, "TwinsCoin: A cryptocurrency via proof-of-work and proof-of-stake," in *Proc. 2nd ACM Workshop Blockchains, Cryptocurrencies, Contracts*. New York, NY, USA: Association for Computing Machinery, May 2018, pp. 1–13, doi: 10.1145/3205230.3205233.

[135] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, pp. 118541–118555, 2019.

[136] Z. Liu, S. Tang, S. S. M. Chow, Z. Liu, and Y. Long, "Fork-free hybrid consensus with flexible proof-of-activity," *Future Gener. Comput. Syst.*, vol. 96, pp. 515–524, Jul. 2019.

[137] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. USENIX Symp. Netw. Syst. Design Implement.*, 2016, pp. 45–59.

[138] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending Bitcoin's proof of work via proof of stake [Extended Abstract]y," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, Dec. 2014.

[139] C. Li and B. Palanisamy, "Comparison of decentralization in DPoS and PoW blockchains," in *Blockchain—ICBC 2020*, Z. Chen, L. Cui, B. Palanisamy, and L.-J. Zhang, Eds. Cham, Switzerland: Springer, Sep. 2020, pp. 18–32.

[140] C. Liu, Y. Xiao, V. Javangula, Q. Hu, S. Wang, and X. Cheng, "NormaChain: A blockchain-based normalized autonomous transaction settlement system for IoT-based E-Commerce," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4680–4693, Jun. 2019.

[141] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," 2017, *arXiv:1708.03778*.

[142] M. Mouly, M.-B. Pautet, and T. Foreword By-Haug, *The GSM System for Mobile Communiations*. Utrecht, The Netherlands: Telecom Publishing, 1992.

[143] N. Arnosti and S. M. Weinberg, "Bitcoin: A natural oligopoly," *Manage. Sci.*, vol. 68, no. 7, pp. 4755–4771, Jul. 2022.

[144] P. Vasin. (2014). *Blackcoins Proof-of-Stake Protocol V2*. [Online]. Available: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf

[145] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, p. 326, Jan. 2019.

[146] C. Liu, K. K. Chai, X. Zhang, and Y. Chen, "Peer-to-peer electricity trading system: Smart contracts based proof-of-benefit consensus protocol," *Wireless Netw.*, vol. 27, no. 6, pp. 4217–4228, Aug. 2021.

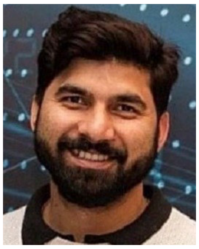
[147] A. Buldas, D. Draheim, M. Gault, R. Laanoja, T. Nagumo, M. Saarepera, S. A. Shah, J. Simm, J. Steiner, T. Tammet, and A. Truu, "An ultra-scalable blockchain platform for universal asset tokenization: Design and implementation," *IEEE Access*, vol. 10, pp. 77284–77322, 2022.

[148] G. Wood. (2016). *Polkadot: Vision for a Heterogenous Multi-Chain Framework, Draft 1*. [Online]. Available: https://polkadot.network/PolkaDotPaper.pdf

[149] W. Zhao, I. M. Aldyaflah, P. Gangwani, S. Joshi, H. Upadhyay, and L. Lagos, "A blockchain-facilitated secure sensing data processing and logging system," *IEEE Access*, vol. 11, pp. 21712–21728, 2023.

[150] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything you wanted to know about the blockchain: Its promise, components, processes, and problems," *IEEE Consum. Electron. Mag.*, vol. 7, no. 4, pp. 6–14, Jul. 2018.

VIPIN DEVAL received the B.Tech. degree in CSE from UPTU, Lucknow, India, and the M.Tech. degree in IT from IIITM, Gwalior, India. He is currently pursuing the Ph.D. degree with Tallinn University of Technology (TalTech), Tallinn, Estonia. In his Ph.D. degree, he is also working on mobile smart contracts to address the scalability of blockchains and efficient consensus mechanisms. He is also an Assistant Professor with the Department of Computer Science and Engineering, KIET Group of Institutions, Ghaziabad, Delhi NCR, India. His research interests include blockchain and smart contracts.



VIMAL KUMAR DWIVEDI received the Ph.D. degree in computer science and engineering from Tallinn University of Technology (TalTech), Tallinn, Estonia. His Ph.D. project was partially sponsored by the Qtum Foundation, Singapore. As a part of their Ph.D. research, he has developed an XML-based smart contract language that allows non-IT blockchain practitioners to write smart contracts for business use cases. He is currently a Research Fellow with the School of Electronics, Electrical Engineering, and Computer Science (EEECS), Queen's University Belfast (QUB), Northern Ireland, U.K., with Prof. Karen and an Asst. Prof. Vishal Sharma on a project titled digital twin and blockchain for business decision modeling. Before coming to QUB, he was a Lecturer with the Institute of Computer Science, University of Tartu, Tartu, Estonia. Prior to this, he was an early-stage Researcher with the Information Systems Group, TalTech. He serves as a guest editor for mathematics and computer science.

ABHISHEK DIXIT received the master's degree in software engineering from Thapar University, India. He is currently pursuing the Ph.D. degree with the Blockchain Technology Group, Tallinn University of Technology (TalTech). He is also an early-stage Researcher with the Blockchain Technology Group, TalTech. He was an Assistant Professor in computer science in India. He has coauthored three publications in conference proceedings. His research interests include human-centered computing and multi-agent technology for blockchains and smart contracts.

ALEX NORTA (Senior Member, IEEE) received the M.Sc. degree from the Johannes Kepler University of Linz, Austria, in 2001, and the Ph.D. degree from the Eindhoven University of Technology, The Netherlands, in 2007. He is currently a blockchain Research Member of Tallinn University, Estonia. He was a Researcher with Oulu University Secure-Programming Group (OUSPG) after having been a Postdoctoral Researcher with the University of Helsinki, Finland. His Ph.D. thesis was partly financed by the IST project CrossWork, in which he focused on developing the eSourcing concept for dynamic inter-organizational business process collaboration. His research interests include business-process collaboration, smart contracts, blockchain technology, e-business transactions, service-oriented computing, software architectures, software engineering, ontologies, security, multi-agent systems, distributed business-intelligence mining, e-learning, Agile software engineering, production automation, enterprise architectures, and e-governance. For the blockchain-tech startups Qtum.org, their respective whitepapers, and also serves as an Advisor for several other blockchain-tech startups, such as Cashaa.

SYED ATTIQUE SHAH (Senior Member, IEEE) received the Ph.D. degree from the Institute of Informatics, Istanbul Technical University, İstanbul, Turkey. During the Ph.D. degree, he studied as a Visiting Scholar with The University of Tokyo, Japan; National Chiao Tung University, Taiwan; and Tallinn University of Technology, Estonia, where he completed the major content of his thesis. He was an Associate Professor and a Chairperson of the Department of Computer Science, BUITEMS, Quetta, Pakistan. He was also engaged as a Lecturer with the Data Systems Group, Institute of Computer Science, University of Tartu, Estonia. Currently, he is a Lecturer in smart computer systems with the School of Computing and Digital Technology, Birmingham City University, U.K. His research interests include big data analytics, the Internet of Things, machine learning, network security, and information management.

RAHUL SHARMA received the Ph.D. degree from Tallinn University of Technology (TalTech), Tallinn, Estonia. Currently, he is a Full Professor and the Head of the Information Technology Department, Ajay Kumar Garg Engineering College, Ghaziabad. Simultaneously, he contributes as a Senior Researcher with TalTech. He is also a Distinguished Scholar. He is recognized for his passionate dedication to research and dynamic leadership in the field.

DIRK DRAHEIM (Member, IEEE) is currently a Full Professor in information systems and the Head of the Information Systems Group, Tallinn University of Technology. He has coauthored over 120 publications in international journals, conference proceedings, and four Springer books. His research interests include the architecture, design, and semantics of large-scale information systems.