

---

# CONSENSUS ALGORITHMS OF DISTRIBUTED LEDGER TECHNOLOGY - A COMPREHENSIVE ANALYSIS

---

A PREPRINT

 **Ahmad J. Alkhodair**  
Computer Engineering Department  
University of Tabuk  
Tabuk, Saudi Arabia  
aalkhodair@ut.edu.sa

 **Saraju P. Mohanty**  
Department of Computer Science and Engineering  
University of North Texas  
Denton, TX 76203, USA  
saraju.mohanty@unt.edu

 **Elias Kougianos**  
Department of Electrical Engineering  
University of North Texas  
Denton, TX 76203, USA  
USA  
elias.kougianos@unt.edu

September 23, 2023

## ABSTRACT

The most essential component of every Distributed Ledger Technology (DLT) is the Consensus Algorithm (CA), which enables users to reach a consensus in a decentralized and distributed manner. Numerous CA exist, but their viability for particular applications varies, making their trade-offs a crucial factor to consider when implementing DLT in a specific field. This article provided a comprehensive analysis of the various consensus algorithms used in distributed ledger technologies (DLT) and blockchain networks. We cover an extensive array of thirty consensus algorithms. Eleven attributes including hardware requirements, pre-trust level, tolerance level, and more, were used to generate a series of comparison tables evaluating these consensus algorithms. In addition, we discuss DLT classifications, the categories of certain consensus algorithms, and provide examples of authentication-focused and data-storage-focused DLTs. In addition, we analyze the pros and cons of particular consensus algorithms, such as Nominated Proof of Stake (NPoS), Bonded Proof of Stake (BPoS), and Avalanche. In conclusion, we discuss the applicability of these consensus algorithms to various Cyber Physical System (CPS) use cases, including supply chain management, intelligent transportation systems, and smart healthcare.

**Keywords** Distributed Ledger Technology (DLT) · Blockchain (BC) · Consensus Algorithm (CA) · Cyber Physical Systems (CPS) · Internet of Things (IoT) · Smart City.

## I. Introduction

Distributed Ledger Technologies (DLTs) have disrupted business paradigms and changed how companies handle data and transactions. DLTs are predicted to reach \$2,082.8 million [1] of transactions. The Quorum blockchain platform streamlines cross-border payments and reduces expenses. DLTs provide secure and efficient medical record sharing, improving patient care and privacy. Walmart and other retail giants use blockchain technology to improve supply chain traceability and product quality. As demand for secure, transparent, and efficient solutions rises, researchers, developers, and industry professionals must understand DLTs' success mechanisms [2].

DLTs are a novel method for administering and exchanging data across a network of participants without a central authority. They provide a secure, transparent, and tamper-resistant method for recording and sharing information,

which has the potential to improve trust, productivity, and collaboration across a variety of applications and industries [3].

The most prominent and extensively adopted form of DLT is the blockchain. It is a decentralized digital ledger that uses cryptographic techniques and consensus algorithms to maintain an immutable chain of blocks, or records. Each block contains a set of transactions, and once a block is added to the chain it becomes practically impossible to alter the information it contains without the agreement of all network participants [4].

Key concepts related to distributed ledger technology and blockchain include the use of cryptographic hashing and consensus mechanism guarantees that the data stored on the blockchain is secure and unchangeable. Once a block is added to the chain, it is extremely difficult to alter its content, which helps preserve the ledger's integrity and credibility [5].

The blockchain eliminates the requirements for a centralized authority to govern and maintain the ledger. Instead, the responsibility for maintaining and updating it is distributed across the network nodes. This decentralized approach improves the system's security, transparency, and resistance to censorship [6].

All network participants can view the transactions recorded on the blockchain, fostering transparency and allowing anyone to audit and verify the data. This feature is especially useful in applications where data integrity and trust are essential, such as financial transactions and supply chain management [7]. Multiple applications of DLTs that use different consensus mechanisms are shown in Fig. 1 [8]. The blockchain, or DLTs in general have been applied to almost all aspects of human life including healthcare, agriculture, pharmaceutical, transportation, and supply chain [9], [10], [11], [12].

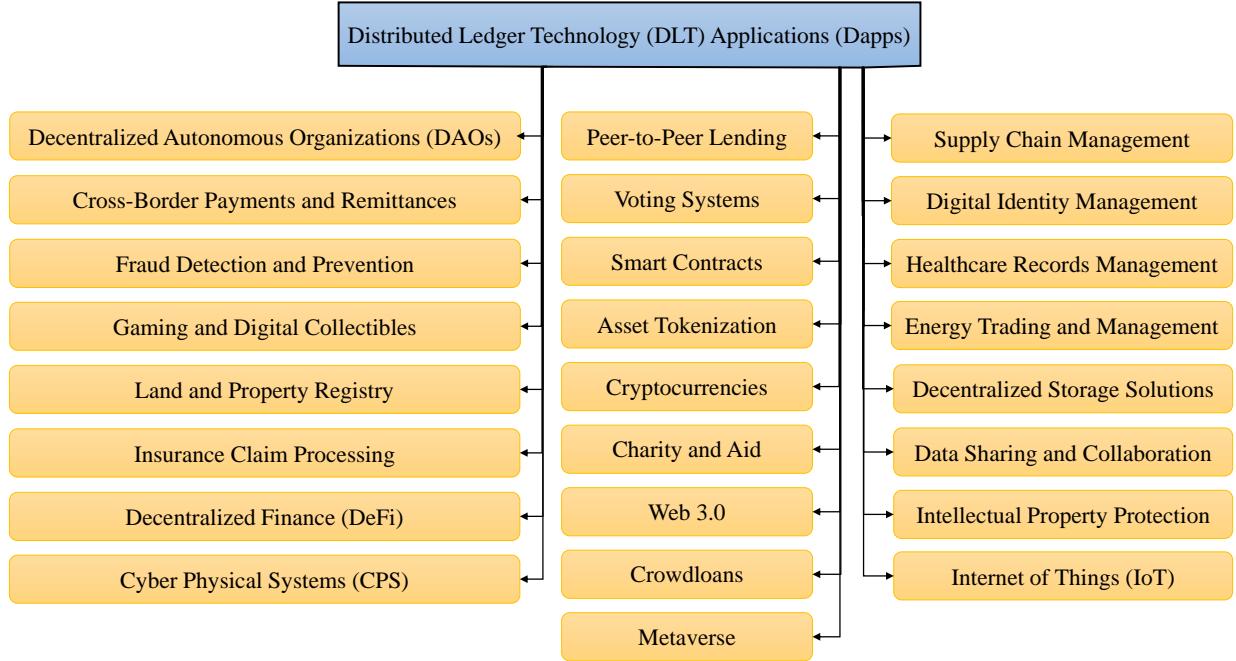


Fig. 1. DLT Applications.

For example, cryptocurrencies are digital or virtual currencies that use encryption to make sure that transactions are safe and that control is not centralized. Supply Chain Management (SCM) is the process of making things and services easier to track, more transparent, and more efficient. Voting systems must be safe, clear, and unchangeable. Smart contracts should automatically put the terms of the contract into effect when certain conditions are met. Decentralized Finance (DeFi) means financial services that are made on decentralized platforms, which eliminate and make financial products easier to get. Digital identity management is the process of making sure that digital identities are safe and can be checked. This improves privacy and speeds up the authentication process. The Internet of Things (IoT) makes it possible for connected devices to share data in a safe and clear way, which improves trust and coordination. Intellectual property protection improves the way copyrights, patents, and trademarks are managed by keeping records that cannot be changed. Healthcare records management is the safe and efficient

storage, sharing, and control of sensitive patient data by all healthcare providers. Energy trading and management decentralized systems involves making, using, and trading energy and support renewable energy and efficient energy markets. Cross-border payments and remittances make foreign transactions faster, cheaper, and safer by cutting out middlemen and limiting fees.

Asset tokenization is the process of turning real or digital assets into digital tokens and putting them in a blockchain. This makes it easier to share ownership and trade. Decentralized Autonomous Organizations (DAOs) are organizations without a central authority or hierarchy that are run by pre-set rules and consensus processes. Data sharing and collaboration data between people, groups, or devices must be done in a way that keeps the data private and secure. Processing insurance claims requires streamlining and automating the way they are handled, limiting fraud, and making the customer experience better. Land and property registry requires keeping clear, safe, and unchangeable records of who owns land and property, which builds trust and reduces disagreements. Decentralized storage solutions are networks of secure, distributed data storage that make data more secure, reliable, and easy to reach. Peer-to-peer lending takes place when people trade and borrow money directly from each other, without any middlemen. This gives people better rates and easier access to credit. Gaming and digital collectibles involves making, selling, and proving ownership of unique digital assets like non-fungible tokens (NFTs). Finally, fraud discovery and prevention demands better security and openness in transactions and data management, which lowers the risk of fraud and other bad things happening. The applications that suit DLT usage increase every day due to the wide variety of ledgers and consensus algorithms.

Consensus algorithms are an essential component of DLT systems, as they play a crucial role in sustaining the distributed ledger's integrity, security, and stability. These algorithms provide a method for network participants to concur on the ledger's validity and state, ensuring that all nodes have a consistent and accurate view of the data. Transaction validation, network synchronization, conflict resolution, and overall system security are the responsibility of consensus mechanisms. A DLT system can be tailored to satisfy specific goals and requirements, such as scalability, energy efficiency, and decentralization, by selecting the appropriate consensus algorithm [13].

### A. Paper Organization

The paper is organized as shown in Fig. 2. The paper organization starts with an introduction in Sec. I that presents the background, motivation, and purpose of the study, as well as an overview of the DLT and its applications. Finally, it presents the significance of consensus algorithms. Sec. II presents firstly the crucial role of consensus algorithms and then moves to the motivation of the paper. Sec. III discusses related works and their contributions. Sec. IV discusses DLT classifications and their consensus mechanisms. Sec. V explores multiple consensus algorithms and their pros and cons. Next, Sec. VI presents the method used to evaluate the consensus algorithms. Sec. VII examines the evaluation and their results. Sec. VIII concludes the work presented in this paper.

## II. Contributions of the Current Paper

In the realm of DLTs, consensus algorithms play a crucial role in ensuring the security, integrity, and consistency of the shared ledger across network participants. A multitude of consensus algorithms have been proposed, each with distinct features and trade-offs. When evaluating these algorithms, it is essential to consider their suitability for specific applications. For instance, Proof of Work (PoW), employed by Bitcoin and, initially, Ethereum, offers robust security but suffers from high energy consumption and limited scalability. In contrast, Proof of Stake (PoS), adopted by Ethereum 2.0 and Cardano, addresses these limitations while maintaining adequate security, making it more suitable for energy-conscious and scalable applications. Delegated Proof of Stake (DPoS), as seen in EOS and Tron, further enhance scalability, but at the cost of centralization. Practical Byzantine Fault Tolerance (PBFT) and its variants, used in permissioned DLT systems like Hyperledger Fabric, offer high transaction throughput and low latency, ideal for enterprise applications. Ultimately, selecting the appropriate consensus algorithm hinges on the specific requirements of the target application, with considerations such as security, scalability, decentralization, and energy efficiency driving the decision-making process.

### A. Motivation

The motivation for studying consensus algorithms in the context of DLTs stems from the wide range of challenges and requirements that different DLT systems encounter. These diverse systems necessitate the development and implementation of various consensus algorithms, each with their unique properties, trade-offs, and performance characteristics. As DLTs are being adopted across numerous industries, it is crucial to have a comprehensive understanding of the underlying consensus mechanisms that govern their operation.

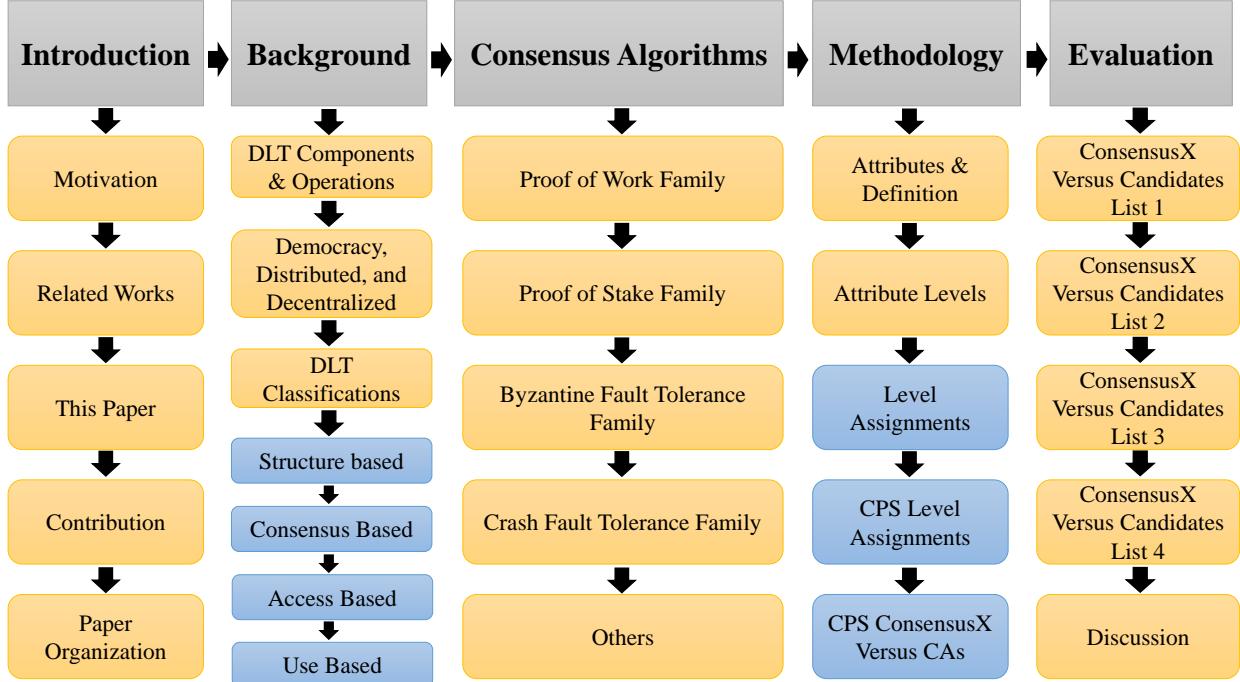


Fig. 2. Paper Organization.

One of the primary challenges faced by DLT systems is scalability, which is as the number of participants and transactions in a DLT system grows, and ensures that the system is able to handle the increased load while maintaining performance and security. Different consensus algorithms address scalability in various ways, affecting the overall efficiency and throughput of the system.

Another challenge is ensuring the security and integrity of the distributed ledger. It is a critical requirement for any DLT system. Consensus algorithms play a vital role in maintaining security by preventing attacks, such as double-spending or Sybil attacks, and ensuring that only valid transactions are added to the ledger.

Decentralization is a key characteristic of many DLT systems, which removes the need for a central authority and distributes control among network participants. Consensus algorithms must strike a balance between decentralization and other factors, such as efficiency and security, to ensure the optimal functioning of the system.

Moreover, energy efficiency is vital. Some consensus mechanisms, such as PoW, consume significant amounts of energy, which has environmental and economic implications. As a result, there is a growing interest in exploring alternative consensus algorithms that are more energy-efficient while maintaining the desired levels of security and performance.

Lastly, latency, which in some applications such as financial transactions or real-time data sharing is crucial in minimizing the amount of time spent in transaction confirmation and data propagation. Consensus algorithms play a significant role in determining the speed at which transactions are validated and added to the ledger.

For researchers, developers, and practitioners working with DLTs, understanding the various consensus algorithms is essential to make informed decisions about which mechanism is best suited for their specific use case. Categorizing and comparing these algorithms based on their properties, trade-offs, and performance allows for a more in-depth analysis and a better understanding of the advantages and limitations of each approach. This knowledge can ultimately lead to the development of more efficient, secure, and scalable DLT systems that cater to the diverse needs of different industries and applications.

## B. Contributions

The main goals and objectives of this article are to provide a comprehensive review of different consensus algorithms by:

- Presenting an in-depth analysis of various consensus mechanisms used in DLTs, highlighting their features, advantages, and limitations. This will help readers gain a better understanding of the diverse consensus algorithms and their applicability to different use cases and requirements.
- Developing a classification scheme for consensus algorithms: Propose a systematic classification of consensus algorithms based on their underlying properties, such as proof-based, voting-based, weight-based, randomized, and Directed Acyclic Graph (DAG)-based mechanisms. This classification will facilitate the comparison and analysis of different consensus mechanisms, making it easier for researchers and practitioners to identify suitable algorithms for their specific needs.
- Investigating different DLT architectures: Examine various DLT architectures, focusing on how data is stored and linked within these systems. This investigation will provide insights into the design choices and trade-offs associated with different DLT architectures, helping readers make informed decisions when selecting or developing DLT solutions.
- Analyzing the relationship between consensus algorithms and DLT architectures: Explore the interplay between consensus mechanisms and DLT architectures, identifying how the choice of consensus algorithm can impact the overall performance, security, and scalability of a DLT system. This analysis will contribute to the understanding of how consensus algorithms and DLT architectures can be optimized to meet the diverse requirements of different industries and applications.
- Contributing to the existing body of knowledge: By offering a comprehensive review, classification, and analysis of consensus algorithms and DLT architectures, this article aims to contribute to the ongoing research and development efforts in the field of DLTs. The insights provided in this article will not only benefit researchers and developers working with DLTs but will also inform decision-makers, policymakers, and other stakeholders interested in the potential applications and implications of these technologies.

By achieving these goals and objectives, this article will add significant value to the existing literature on consensus algorithms and DLTs, facilitating a deeper understanding of these technologies and their potential impact on various industries and applications.

### III. Related Prior Works

We have reviewed prior works contributing to various type of surveys and overview works on blockchain and DLT. Table I compares the contribution of each related work to the current paper.

The work in [14] provides an extensive overview of various consensus algorithms used in blockchain technology, focusing on their strengths and weaknesses. The authors analyze each algorithm's trade-offs in terms of decentralization, security, and performance, ultimately highlighting the ongoing challenges in achieving a balance between these factors.

The work [15] investigates the Bitcoin Unlimited mining protocol, emphasizing the importance of a prescribed block validity consensus. Through their analysis, the authors reveal potential issues with Bitcoin Unlimited, including risks of blockchain forks, which can compromise the security and stability of the network.

Authors in [16] examine the scalability challenges faced by blockchain technology, comparing Proof-of-Work (PoW) and Byzantine Fault Tolerance (BFT) replication mechanisms. The author highlights the trade-offs between decentralization, security, and performance, suggesting that a combination of PoW and BFT might offer a more scalable and secure solution for blockchain systems.

The work in [17] provides a comparative study of various blockchain consensus algorithms, focusing on their strengths, weaknesses, and applicability in different scenarios. The authors highlight the importance of selecting the appropriate consensus algorithm based on specific use cases and system requirements, emphasizing the trade-offs among security, performance, and decentralization.

A comprehensive overview of blockchain technology in their NIST report, focusing on the underlying concepts, consensus mechanisms, and various use cases has been presented in [18]. The authors discuss the challenges and limitations of current blockchain implementations, offering guidance for organizations interested in adopting or developing blockchain-based systems.

The work [19] provides a systematic survey of various consensus algorithms employed in blockchain technology. The authors discuss the strengths and weaknesses of each algorithm, highlighting their suitability for different application scenarios. The survey offers valuable insights for researchers and practitioners exploring consensus mechanisms for their blockchain projects.

The work in [20] presents an extensive survey on consensus algorithms utilized in blockchain systems, discussing their properties and functions. The authors compare the algorithms based on performance, security, and decentralization, offering a valuable resource for understanding the trade-offs and selection criteria for different blockchain applications.

The work [21] conducts a comparative analysis of blockchain consensus algorithms, evaluating their performance, security, and decentralization properties. The authors emphasize the importance of selecting the right consensus mechanism for specific use cases, and their study serves as a helpful guide for practitioners and researchers in the field of blockchain technology.

A comprehensive survey of consensus mechanisms in blockchain technology, specifically focusing on applications in IoT networks has been provided in [22]. The authors analyze the properties and requirements of each consensus algorithm, offering valuable insights into their suitability for various IoT use cases and challenges.

The work in [23] provides a thorough review of blockchain consensus algorithms, discussing their properties, advantages, and disadvantages. The authors examine various consensus mechanisms in the context of diverse application scenarios, offering a valuable resource for researchers and practitioners seeking to choose the right algorithm for their blockchain projects.

The authors in [24] perform a comparative analysis and classification of consensus algorithms in blockchain technology. The authors evaluate each mechanism based on key characteristics, providing a systematic overview that helps readers understand their strengths and weaknesses, as well as their suitability for different application domains.

The work in [25] conducts a comprehensive study of blockchain consensus algorithms, examining their features, security aspects, and performance. The authors delve into various consensus mechanisms, offering a detailed comparison that aids researchers and practitioners in understanding the trade-offs and selecting appropriate algorithms for their blockchain projects.

The work [26] provides a survey of different types of consensus algorithms used in blockchain networks and discusses their strengths and weaknesses. It also describes how consensus algorithms are integrated with other technologies, such as BFT (Byzantine Fault Tolerance), credit mechanisms, and artificial intelligence algorithms. The paper provides a detailed analysis of three types of consensus algorithms: those based on certain attribute value proof of peers, those based on peer voting mechanism, and Paxos class consensus algorithm. Additionally, the paper conducts comparative research according to Munde ll's impossible triangle theory and gives the development direction of consensus algorithm. Overall, the paper provides a useful reference for researchers to conduct in-depth research on blockchain consensus algorithms.

In [27], a survey and analysis of different consensus algorithms used in Blockchain based on Post Quantum Cryptosystems has been presented. It explains that in a decentralized network, trust among the nodes is important and consensus algorithms are used to ensure that every node agrees before any data is added to the network. The paper also discusses how quantum computing can impact the security of Blockchain and proposes solutions to make Blockchain systems quantum safe. Overall, the paper provides a comprehensive overview of consensus algorithms in Blockchain and their relationship with post-quantum cryptography.

In [28], consensus mechanisms for electricity information acquisition systems has been discussed. It analyzes the problems faced by the current power system data sharing and proposes three consensus algorithms that are suitable for solving these problems. The paper compares and improves these algorithms to make them more efficient and energy-saving. The proposed solutions have the potential to improve data sharing in the power system, increase efficiency, and reduce energy consumption. Overall, this paper provides valuable insights into the challenges faced by the current power system data sharing and proposes efficient solutions to overcome them.

The work in [29] presents a comprehensive survey of the latest consensus mechanisms used in blockchain systems, along with their advantages and disadvantages. It systematically studies the state-of-the-art consensus mechanisms, and looks into their pros and cons, respectively. The two most widely used consensus mechanisms are Proof of Work (PoW) and Proof of Stake (PoS), but there are also other consensus algorithms that derive from either method such as Delegated Proof of Stake (DPoS). The paper compares these algorithms from multiple perspectives including security, fault tolerance, validation speed, energy consumption, and level of decentralization. It also takes a closer look at the different problems each algorithm has to deal with.

The work in [30] contains a comprehensive survey of consensus algorithms in blockchain-based applications. It presents a taxonomy of consensus algorithms, including proof-based and voting-based algorithms, and provides a comparative discussion of their performances, efficiencies, and uses in blockchains. The paper also analyzes the application domains of consensus algorithms in terms of development tools, uses, and environments. Additionally, it highlights the challenges in blockchain applications regarding functional and non-functional issues. Overall, this paper

provides valuable insights for stakeholders interested in understanding the trends and issues related to consensus algorithms in blockchain technology.

In [31], the application of blockchain in digital rights protection, with a focus on the improvements made by researchers to the consensus algorithm has been presented. The consensus algorithm is the core of blockchain technology, but it cannot be directly applied to digital rights protection due to its drawbacks. Therefore, researchers have optimized and improved the consensus algorithm to meet the needs of digital rights protection. The paper reviews these improvements and analyzes their impact on digital rights protection.

In [32], a survey on Decentralized Consensus Mechanisms for Cyber Physical Systems has been presented. The authors discuss the challenges of deploying decentralized consensus protocols to ensure fairness, trust, and transparency in operations. They also present an evaluation model of consensus approaches based on defined metrics and categorize them through parameters such as incentive, performance, data model, energy-efficiency, and exposure likelihood.

A comprehensive review of consensus algorithms in blockchain technology has been presented in [33]. It proposes a multi-dimensional tradeoff model to guide the construction of consensus algorithms, taking into account the technical constraints that limit the large-scale application of blockchain, namely scalability, security, and decentralization. The paper compares and analyzes various classical consensus algorithms and focuses on their design principles under the multi-dimensional tradeoff model. It also describes the performance indicators of different consensus algorithms and provides different solutions for blockchain in different dimensions. Overall, this paper offers operational advice to developers and users on how to design optimal consensus algorithms for their specific needs.

The work in [34] surveys the consensus algorithms used in mobile healthcare blockchain networks. It discusses the importance of consensus algorithms in blockchain distributed ledgers and provides guidance on selecting the most suitable consensus algorithm for a given network. The paper also presents a theoretical approach to improving existing blockchain structures in mobile healthcare settings. However, it does not provide specific details on the proposed approach or its potential limitations.

## IV. Background: Distributed Ledger Technology (DLT)

One of the most revolutionary developments in recent years has been the introduction of blockchain and Distributed Ledger Technologies (DLTs), which have altered many facets of our life. These innovative methods of data storage and transaction processing may cause widespread change across a variety of sectors, from banking and supply chain management to government and healthcare. Yet, it is essential to note that blockchain technology and DLT are not the same thing.

The blockchain is a sort of distributed ledger technology (DLT) that uses a distributed network of computers to record transactions in a way that is both public and immutable. Blocks of transactions are linked using cryptographic hashes to create a linear and immutable chain. Because of how this system is set up, it is impossible to make illegal changes or spend money twice. Bitcoin, a digital currency that facilitates peer-to-peer transactions without a central authority, is the most well-known application of blockchain technology.

In contrast, distributed ledger technology is a catch-all phrase for any digital system in which numerous nodes work together to keep a single, authoritative record of all transactions. While blockchains do fall under the umbrella term of distributed ledger technology (DLT), not all DLTs are blockchains. There are DLTs that don't rely on a sequential chaining of blocks but instead employ different data structures and consensus procedures. The primary feature of DLTs is their decentralized nature, which improves security, fosters transparency, and minimizes the need for a centralized authority.

Blockchain and other forms of distributed ledger technologies rely heavily on consensus mechanisms. To keep the ledger honest and consistent, these algorithms make sure that every node in the network has the same information.

Table II outlines the differences between Distributed Ledger Technology (DLT) and Blockchain technology.

### A. Distributed Ledger Technology Versus Blockchain Technology

DLT and the blockchain are related. Both systems use a distributed network of nodes to manage a shared digital ledger, but they differ significantly. The main differences between blockchain and distributed ledger technologies are as follows:

- 1) Data structures distinguish distributed ledger and blockchain technologies. Blockchain uses cryptographic hashed blocks added consecutively to a chain to record and verify financial transactions. This approach makes

TABLE I  
RELATED WORKS.

Work	Description
Bano et al. [2017] [14]	Assessment of algorithm's decentralization, security, and performance trade-offs, stressing the continuous difficulties in balancing these variables.
Xin et al. [2018] [17]	The authors emphasize security, performance, and decentralization trade-offs when choosing a consensus method based on use cases and system needs.
Yaga et al. [2018] [18]	A discussion on the challenges and limitations of current blockchain implementations, offering guidance for organizations interested in adopting or developing blockchain-based systems.
Wang and Wang [2019] [19]	The authors discuss the strengths and weaknesses of each algorithm, highlighting their suitability for different application scenarios.
Nguyen et al. [2019] [20]	The authors compare the algorithms based on performance, security, and decentralization, offering a valuable resource for understanding the trade-offs and selection criteria for different blockchain applications.
Merkle and Niebling [2019] [21]	Compare blockchain consensus algorithms for performance, security, and decentralization. Their work helps blockchain practitioners and researchers choose the optimal consensus mechanism for certain use cases.
Abbasi et al. [2021] [22]	Give a complete survey of blockchain consensus algorithms for IoT networks. The authors investigate each consensus algorithm's features and needs, providing useful insights into their applicability for specific IoT use cases and issues.
Garg and Choudhary [2021] [23]	The authors examine various consensus mechanisms in the context of diverse application scenarios, offering a valuable resource for researchers and practitioners seeking to choose the right algorithm for their blockchain projects.
Pustokhina et al. [2020] [24]	The authors analyze each mechanism based on important criteria, giving readers a comprehensive overview of their strengths and shortcomings and applicability for different application fields.
Soleymani and Anjomshoa [2020] [25]	The authors delve into various consensus mechanisms, offering a detailed comparison that aids researchers and practitioners in understanding the trade-offs and selecting appropriate algorithms for their blockchain projects.
Deng et al. [2022] [26]	The study analyzes peer voting, attribute value proof, and Paxos class consensus algorithms.
Jose and V [2022] [27]	Analyzes Post Quantum Cryptosystem-based Blockchain consensus techniques.
Ju et al. [2021] [28]	The authors investigate power system data sharing issues and provides three consensus algorithms to address them. The paper compares and optimizes several techniques for conserving energy.
Gu et al. [2021] [29]	The study compares various algorithms on security, fault tolerance, validation speed, energy consumption, and decentralization and examines each algorithm's issues.
Islam et al. [2023] [30]	The authors compares proof-based and voting-based consensus algorithms for performance, efficiency, and blockchain applications and explores consensus algorithm development tools, uses, and contexts.
Guichun et al. [2020] [31]	Examines blockchain's use in digital rights protection and researchers' consensus algorithm upgrades. The study examines improvements and their effects on digital rights protection.
Bodkhe et al. [2020] [32]	Decentralized Consensus Mechanisms for CPS for fairness, trust, and transparency in operations present obstacles and evaluate consensus approaches using motivation, performance, data model, energy-efficiency, and likelihood.
Li et al. [2022] [33]	Authors use a multi-dimensional tradeoff model to guide consensus algorithm construction, considering technical restrictions like scalability, security, and decentralization that limit large-scale blockchain deployment.
Rwibasira and Suchithra [2020] [34]	Authors examine blockchain consensus methods and how to choose the best one for a network. The paper also proposes upgrading mobile healthcare blockchain systems theoretically.
<b>Current Work</b>	Explores 30 consensus algorithms types and their ledger structure, evaluate them based on 11 attributes, and test their suitability to CPS applications.

TABLE II  
DLT VERSUS BC.

Attribute	Distributed Ledger Technology (DLT)	Blockchain Technology
Data Structure	Various structures, such as directed acyclic graphs (DAGs)	Linear chain of blocks linked with hashes
Consensus Mechanisms	Multiple options, e.g., PBFT, DPoS, Federated Consensus	Commonly PoW or PoS
Degree of Decentralization	Varies; can include partially decentralized models	Typically highly decentralized
Scalability & Performance	Potentially higher throughput and performance	Can face scalability challenges
Privacy & Confidentiality	More flexible; can support permissioned or private channels	Transparent and publicly accessible ledgers
Use Cases & Applications	Broad range of industries, e.g., supply chain, healthcare, governance	Often associated with cryptocurrencies & DeFi

transactions public and immutable. DLTs can leverage data structures like DAGs instead of just chaining blocks.

- 2) Distributed ledger technology and blockchain require consensus methods, although their implementations vary. Blockchain consensus methods include PoW and PoS. DLTs can use Practical Byzantine Fault Tolerance (PBFT), Delegated Proof of Stake (DPoS), and Federated Consensus.
- 3) DLT and blockchain are decentralized, although to different degrees. In hybrid decentralized/centralized distributed ledger technology, only a fraction of nodes are trusted to validate transactions and update the ledger. In a blockchain network, any node can contribute to the validation process.
- 4) DLTs' data architecture and consensus processes may outperform blockchains in scalability and performance. DLTs based on DAGs or other data structures can run transactions concurrently to boost transaction throughput. Blockchain systems' linear block structure and resource-intensive consensus mechanisms like PoW make scaling and optimization difficult.
- 5) DLTs may protect user data better than public blockchains. Permissioned DLTs with private channels keep transaction details private by limiting access to the parties involved. Public blockchains provide transaction records.
- 6) DLT and blockchain have the potential to disrupt numerous industries, but their distinctions make them more suited to certain use cases. Blockchain technology is often used for cryptocurrencies and DeFi (Decentralized Finance). DLTs can be customized for supply chain management, healthcare, government, and governance because of their flexible data formats and consensus methods.
- 7) DLT and blockchain both use distributed networks to store shared digital ledgers, but they differ in data structures, consensus mechanisms, degree of decentralization, scalability, privacy, and application space. Understanding these distinctions is crucial for using the best technology in every context.

## B. DLT Components

DLT is an umbrella term that encompasses multiple technologies that enable secure, transparent, and decentralized record-keeping across a network of participants. To comprehend DLT from a top-down perspective, it is necessary to deconstruct and build upon its fundamental components. A step-by-step operational analysis of DLT, beginning with its smallest component and progressing to its most obvious parts follows:

**Cryptography:** The use of cryptographic algorithms to assure data security, integrity, and authentication is the foundation of DLT. Hashing, digital signatures, and public-private key cryptography are crucial for protecting data and facilitating communication between network participants without requiring trust.

**Transactions:** Transactions are the fundamental data element in a distributed ledger. These operations include value transfers, record updates, and the execution of smart contracts. The creators of transactions digitally sign them and distribute them to the network for validation and processing.

**Consensus Mechanisms:** DLT systems rely on consensus algorithms to maintain a secure and consistent ledger, permitting network participants to concur on the validity of transactions. Proof of Work (PoW), Proof of Stake (PoS), and Byzantine Fault Tolerance (BFT) are popular consensus mechanisms. Each mechanism has tradeoffs in areas such as security, efficiency, and resource usage.

**Ledger Structure:** The ledger in a DLT is a data structure that records transactions and maintains an exhaustive, verifiable record of all network activities. The ledger structure can take numerous forms, such as a linear blockchain such as Bitcoin, and Ethereum or a Directed Acyclic Graph (DAG) such as IOTA, depending on the specific DLT implementation.

**Nodes & Network Architecture:** In a DLT, nodes (computers) validate transactions, store the ledger, and communicate with one another in order to maintain the ledger. Full nodes (which store the entire ledger) and lightweight nodes (which store only a portion of the ledger) can have various roles and responsibilities. The architecture of the network is distributed and decentralized, with no singular point of failure or control.

**DLT Applications:** DLT has numerous applications across a wide range of industries and use cases, including finance (cryptocurrencies, remittance, asset tokenization), supply chain management (provenance tracking, inventory control), healthcare (secure data sharing, patient records), and identity management (digital identity, access control).

A top-down approach for understanding DLT entails analyzing its fundamental components, ranging from cryptographic methods and transactions to consensus mechanisms, ledger structures, network architecture, and prospective applications. This all-encompassing perspective clarifies the diverse layers and components that converge to form a secure, transparent, and decentralized system for record-keeping and data management.

Figures 3, 4, 5, and 6 illustrate BC and DLT components at a general high level description.

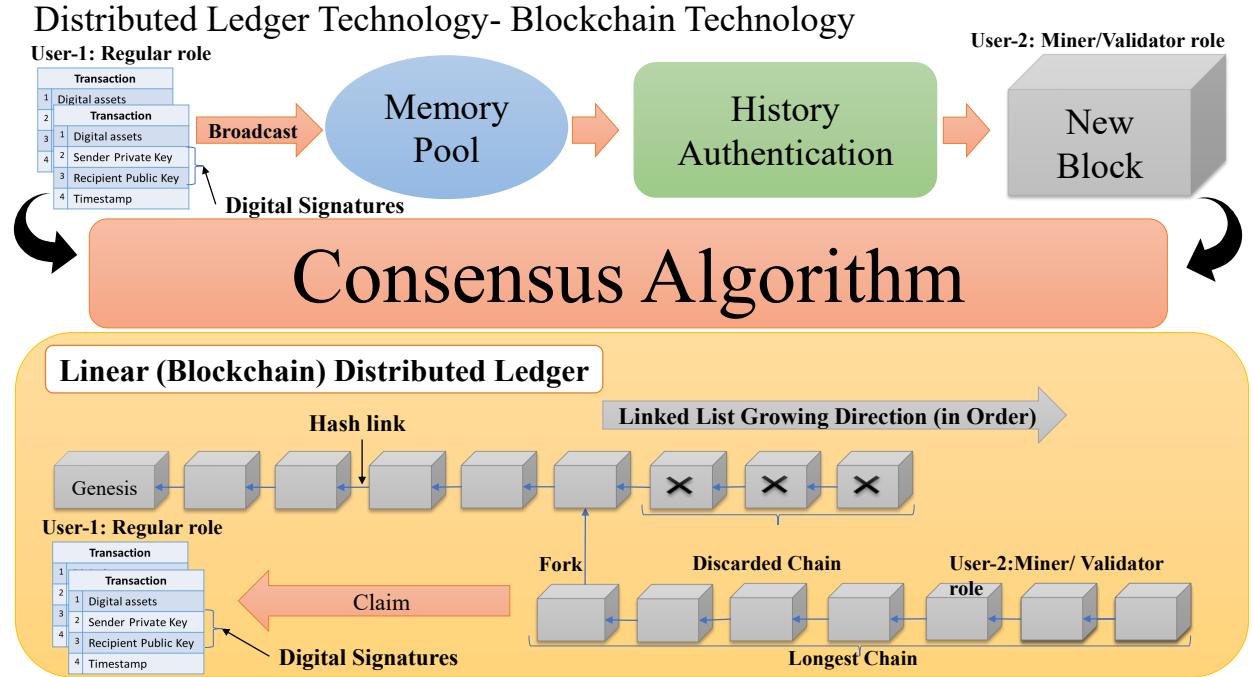


Fig. 3. DLT-Blockchain Depiction.

### C. Consensus Operations

**Mining:** Mining is typically associated with Proof of Work (PoW) consensus algorithms such as Bitcoin's and IOTA's networks. To add a new block of transactions to the blockchain, it is required to solve sophisticated cryptographic puzzles. The first miner to successfully solve these riddles is rewarded with newly created tokens (such as Bitcoins) and transaction fees. Mining contributes to the network's security and decentralization by making it computationally costly to initiate attacks or manipulate the blockchain. Similarly, To add a new transaction to DAG, it is required to solve a lightweight puzzles in IOTA.

**Validation:** Validation is the procedure of confirming the accuracy of blocks and transactions in a DLT network. Validators verify that transactions conform to the network's regulations, including verifying digital signatures, ensuring inputs and outputs are consistent, and preventing double-spending. Once validators have determined that a

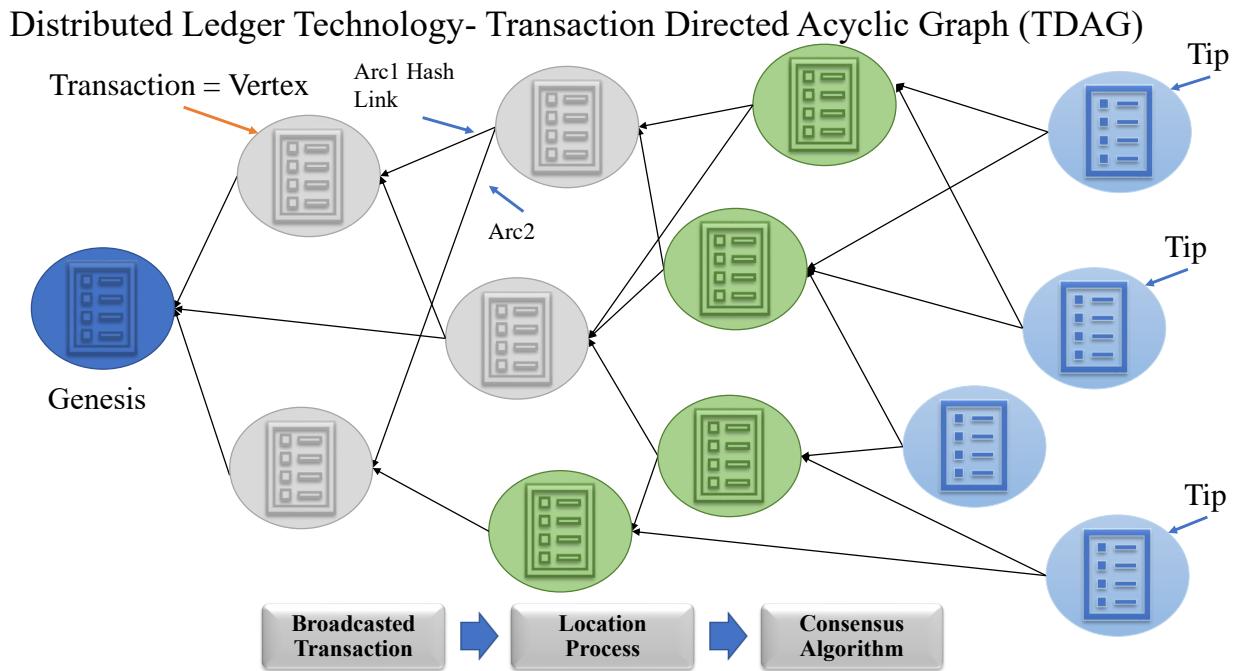


Fig. 4. DLT-Transaction Directed Acyclic Graph (TDAG) Depiction.

### Distributed Ledger Technology- Block Directed Acyclic Graph (BDAG)

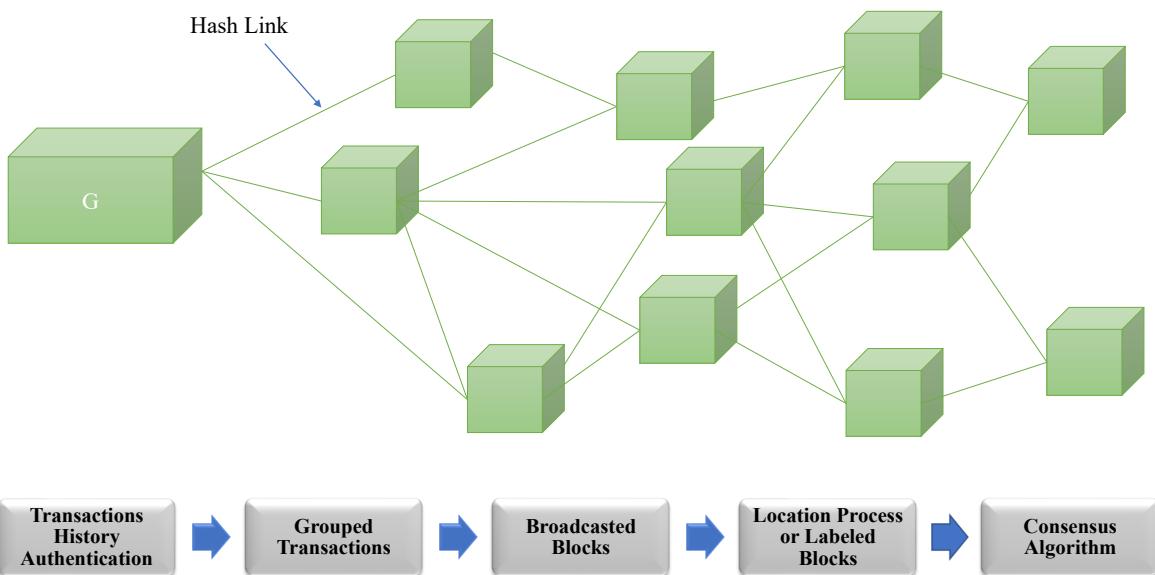


Fig. 5. DLT-Block Directed Acyclic Graph (BDAG) Depiction.

## Distributed Ledger Technology- Multitree Transactions/ Block Directed Acyclic Graph (MDAG)

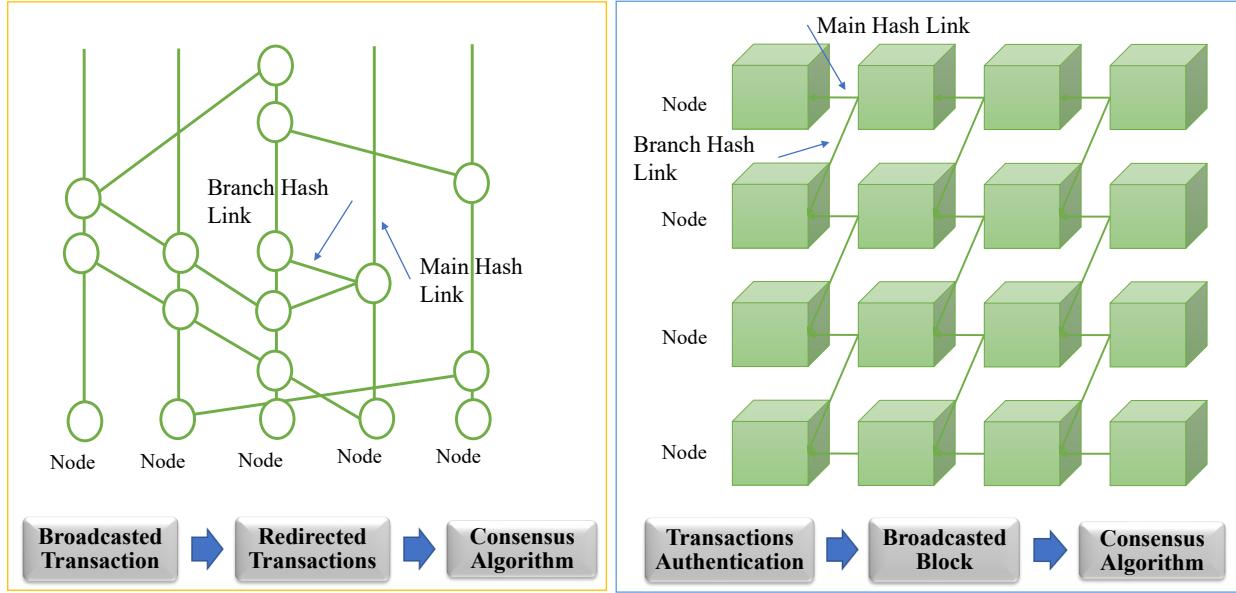


Fig. 6. DLT-Multitree Directed Acyclic Graph (MDAG) Depiction Depiction.

transaction is legitimate, it is added to the blockchain via a new block. Selecting validators and deciding which transactions to include in a block are performed differently by distinct consensus algorithms.

**Voting:** Some consensus algorithms, such as Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT), use voting to reach consensus on the network's state. In these algorithms, a group of validators or delegates are selected, and they deliberate on the legitimacy of transactions and the order in which they should be added to the blockchain. The network reaches consensus when the majority of validators (or a predefined threshold) concur on the same state. Voting helps preserve the network's decentralized nature, provides fault tolerance, and ensures that no singular entity controls the network.

**Authentication:** Authentication in DLT is the procedure of confirming the identity of network participants. This is accomplished using cryptographic techniques, such as public-key cryptography, in which each participant possesses a pair of keys—a private key and a public key. The private key is utilized to sign transactions, whereas the public key is utilized to verify the signature. This enables the network to confirm that transactions are legitimate and authorized by the rightful proprietor of the digital assets or accounts involved. In certain DLT systems, such as permissioned blockchains or private DLTs, authentication also entails verifying that a participant is authorized to access the network, typically through the use of additional identity management systems or access control mechanisms.

### D. Understanding Democracy, Distributed, and Decentralized Concepts in Blockchain and Distributed Ledger Technology

In the context of blockchain technology or distributed ledger technology (DLT), the terms democracy, distributed, and decentralized are often used to describe various aspects of the technology's design and governance. Here's an explanation of each term and how they relate to blockchain and DLT:

**Democracy:** This term refers to the decision-making process within a blockchain network, which can be either centralized or decentralized. In a democratic blockchain, decisions are made collectively by the network's participants, often through voting mechanisms or consensus protocols. This ensures that no single entity has control over the network and its development, fostering a fair and transparent system. Examples of democratic governance models in blockchain include Decentralized Autonomous Organizations (DAOs) and on-chain governance.

**Distributed:** A distributed system refers to a network that is spread across multiple computers or nodes, which work together to maintain and update the shared ledger. In a distributed ledger, each node maintains a copy of the entire ledger, which is constantly updated and synchronized with the rest of the network. This distribution of responsibility provides increased resilience against failures or attacks, as there is no single point of failure. Both blockchain and other forms of DLT are distributed systems by design.

**Decentralized:** Decentralization is a key characteristic of blockchain and DLT, which eliminates the need for a central authority to control and maintain the ledger. In a decentralized system, control is dispersed among the network participants, with no single entity having the power to make unilateral decisions. Decentralization enhances the security, transparency, and censorship-resistance of the system. It also helps to prevent single points of failure and reduces the risk of fraud or manipulation.

The terms democracy, distributed, and decentralized are all important aspects of blockchain technology and DLT. They describe the decision-making processes, network architecture, and control mechanisms that underpin these innovative systems, ensuring their security, transparency, and reliability.

Figure 7 depicts the concepts of Democracy, Distribution, and Decentralization in the context of DLT.

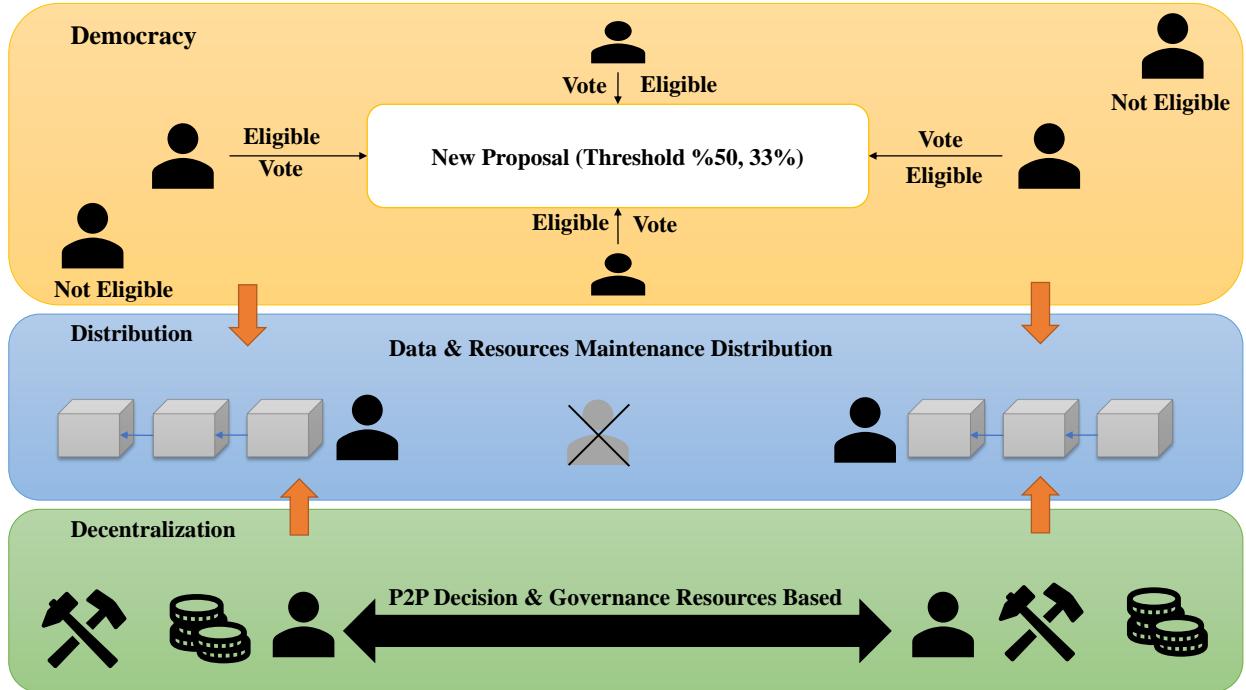


Fig. 7. Democracy, Distribution, Decentralization Concepts.

## E. DLT Classifications

DLT can be classified into various types based on their structure, consensus mechanisms, access control, and use cases. Here is a general classification scheme:

## F. Structure-based Classification

### 1) Linear DLT

Ledgers with a linear data structure, such as Blockchains (BCs), where data is organized into blocks and linked sequentially. Examples include Bitcoin and Ethereum. Distributed Hash Tables (DHTs), where data is organized into transactions and linked sequentially. An example is Holochain (Gossip Protocol).

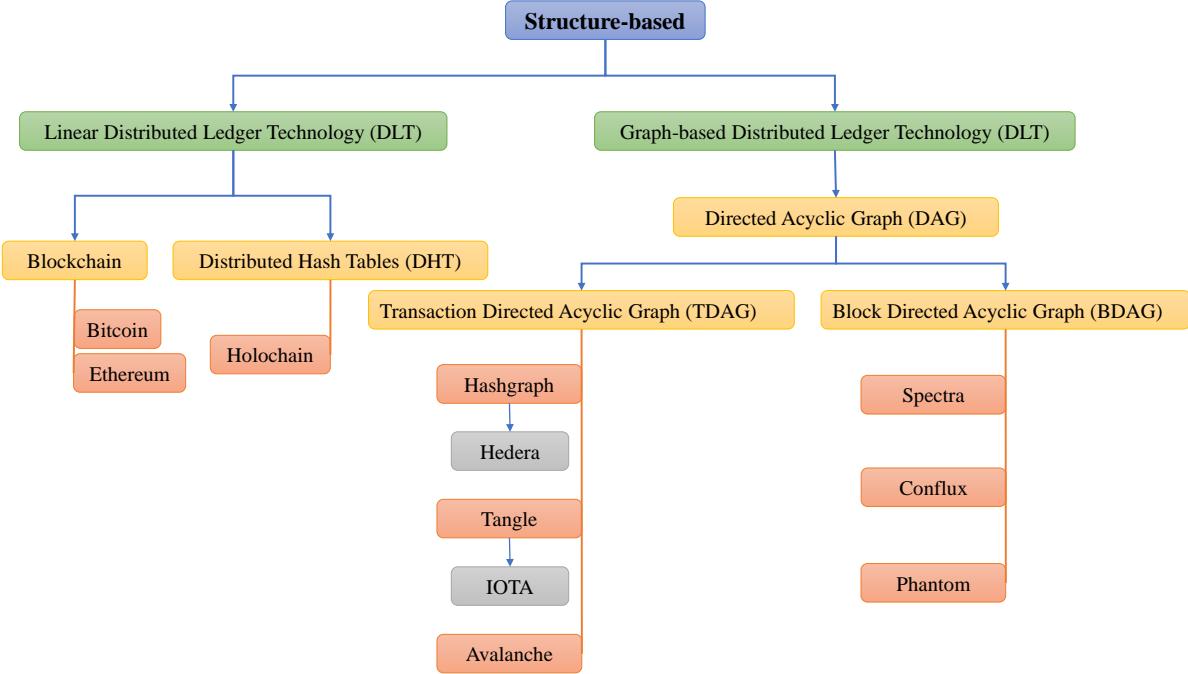


Fig. 8. Structure-Based Classification.

## 2) Graph-based DLT

Ledgers with a graph data structure, such as Directed Acyclic Graphs (DAGs), where data is organized into nodes or vertices and linked in a non-linear, non-circular manner. Examples include IOTA's Tangle and Hedera Hashgraph.

Figure 8 presents the most common structures used in DLT.

## G. Consensus Mechanism-based Classification

### 1) Proof-based

Proof-based consensus algorithms are a category of consensus mechanisms that rely on participants providing some form of proof or evidence to gain the right to create new blocks or validate transactions. The objective of proof-based algorithms is to prevent malicious activities, such as double-spending or Sybil attacks, and ensure the security and reliability of the network. Proof-based algorithms require participants to demonstrate they have made a commitment or invested resources to be eligible to perform specific tasks in the system.

Such schemes include Proof of Work (PoW), Proof of Stake (PoS), Leased Proof of Stake (LPoS), Proof of Elapsed Time (PoET), Proof of Burn (PoB), Proof of Capacity (PoC), Proof of Activity (PoA), Proof of Space (PoSpace), and Proof of Coverage (PoC).

### 2) Voting-based

Voting-based consensus algorithms are a category of consensus mechanisms in which the participants in a distributed system reach an agreement through a process that involves exchanging messages, proposing blocks, and voting on the validity of those blocks. The objective of voting-based algorithms is to achieve consensus in a decentralized manner, despite the possibility of faults, delays, or malicious actions from some participants. In voting-based consensus algorithms, nodes typically follow a set of predefined rules to propose, validate, and vote on transactions or blocks. The proposed blocks or transactions are considered valid and added to the blockchain or distributed ledger when a sufficient number of votes (usually a supermajority) have been collected, indicating that the network agrees on the state of the system.

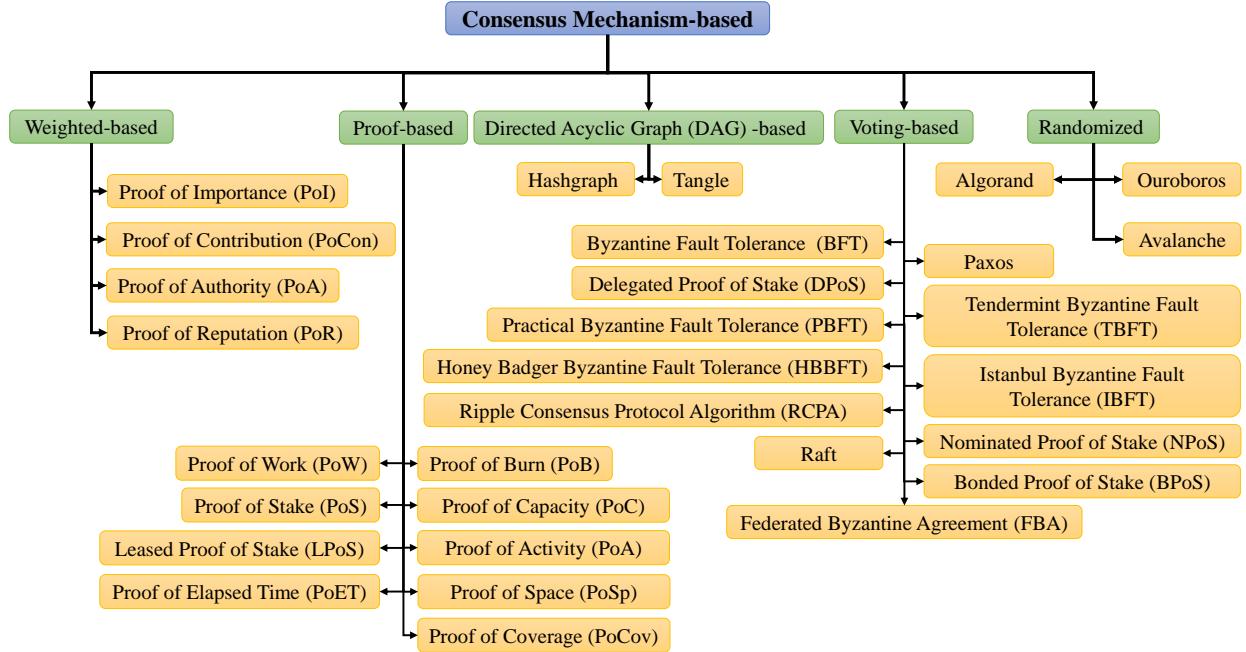


Fig. 9. Consensus-Based Classification.

Examples are Delegated Proof of Stake (DPoS), Byzantine Fault Tolerance (BFT), Practical Byzantine Fault Tolerance (PBFT), Federated Byzantine Agreement (FBA), Ripple Consensus Protocol Algorithm (RCPA), Raft, Paxos, Honey Badger BFT, Tendermint BFT, Istanbul BFT, Nominated Proof of Stake (NPoS), and Bonded Proof of Stake (BPoS).

### 3) Weight-based

Weighted-based consensus algorithms belong to a category of consensus mechanisms in which the influence or decision-making power of the participants in a distributed system is determined by a specific attributes or weights associated with them. This weight can be derived from various factors, such as the amount of cryptocurrency held, the reputation of the participant, or their contribution to the network. The objective of weight-based algorithms is to ensure that the consensus process is fair and resistant to manipulation, while also incentivizing desirable behavior among participants. In weight-based consensus algorithms, the likelihood of a participant being selected to create a new block, validate transactions, or have a higher voting power is proportional to their weight in the system. This mechanism helps to prevent attacks or manipulation by making it more difficult for malicious participants to control the network.

Such weight-based schemes include Proof of Authority (PoA), Proof of Importance (PoI), Proof of Reputation (PoR), and Proof of Contribution (PoC).

### 4) Randomized

Randomized consensus algorithms are a category of consensus mechanisms in which the selection of participants responsible for creating new blocks or validating transactions is based on a random process. These algorithms aim to provide a fair and unbiased mechanism for participant selection, ensuring the security and decentralization of the network. Randomized consensus algorithms often incorporate cryptographic techniques, such as verifiable random functions (VRFs) or cryptographic sortition, to generate random numbers or sequences that determine the selection of block proposers or validators. The randomization process helps to prevent manipulation and collusion among participants, as the chances of being selected are unpredictable and not directly influenced by factors such as computational power, stake, or reputation.

Examples include Randomized PoS (Ouroboros), Randomized Pure PoS (PPoS) (Algorand), and Avalanche.

## 5) DAG-based

DAG-based (Directed Acyclic Graph) consensus algorithms are a category of consensus mechanisms that use a graph-based data structure instead of a linear chain of blocks, as seen in traditional blockchain systems. In a DAG-based system, transactions or data are represented as vertices in the graph, and the edges represent the relationship between them. This data structure allows for multiple transactions to be linked and processed concurrently, enabling higher scalability and faster transaction throughput. DAG-based consensus algorithms aim to overcome some of the limitations of traditional blockchain systems, such as the need for global consensus on a single chain and the associated inefficiencies that arise from this requirement. In DAG-based systems, local consensus can be achieved, and the overall network can maintain its integrity without requiring every node to agree on the state of the entire ledger.

Tangle (PoW), and Hashgraph (BFT) (Gossip about Gossip Protocol) are common examples.

Figure 9 presents the most common consensus algorithms used in DLT.

## H. Access Control-based Classification

**Public DLT:** Open, permissionless networks where anyone can join, participate, and validate transactions. Examples include Bitcoin and Ethereum. **Private DLT:** Restricted, permissioned networks where access is granted only to authorized participants, often used for enterprise solutions. Examples include Hyperledger Fabric and R3 Corda. **Consortium DLT:** Semi-private networks where a group of organizations jointly governs and controls the network. Examples include Quorum and B3i [35] [36] [36].

Figure 10 list the types of access control in DLT.

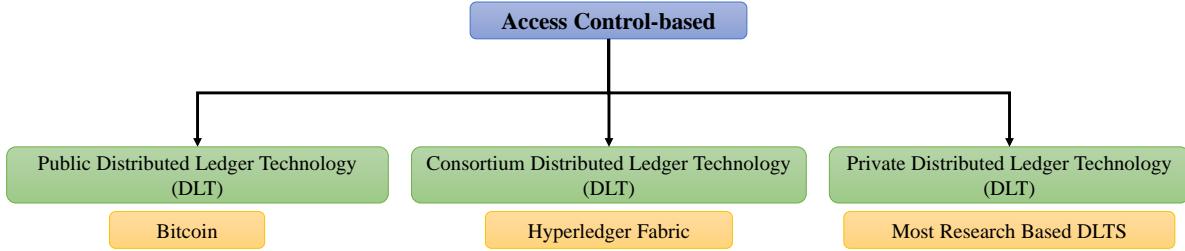


Fig. 10. Access-Based Classification.

## I. Use Case-based Classification

**Cryptocurrency:** DLTs primarily focused on facilitating cryptocurrency transactions. Examples include Bitcoin, Litecoin, and Monero.

**Smart Contract:** DLTs designed to enable the deployment and execution of smart contracts. Examples include Ethereum, Cardano, and Tezos.

**Data Storage:** DLTs designed for decentralized data storage and sharing. Examples include Filecoin, Storj, and Sia.

**Identity and Authentication:** DLTs built for managing digital identities and authentication. Examples include Sovrin and Civic. Presentes in [37] and [38] a novel consensus based in private network operates and distributed on predefined trusted edge nodes.

**Supply Chain Management:** DLTs tailored for improving supply chain transparency and traceability. Examples include VeChain and Waltonchain. In [39], as an example of Decentralized Supply Chain Management (DSCM) in research, the authors used a particular consensus algorithm (PoA) which is known for its suitability to SCM.

**Smart Healthcare:** A decentralized, distributed, intelligent healthcare system eliminates the sole point of failure and third-party control of healthcare data. This means that users have greater control over their medical data and can rest easy knowing that it is secure and confidential. As n research example [40] addresses the problem of single

points of failure (SPoF) in healthcare systems by proposing a decentralized distributed smart healthcare system that eliminates the SPoF and third-party control over healthcare data.

These classifications provide a high-level overview of different DLT types and can help in understanding the various aspects of these technologies. Some DLTs may fall under multiple categories, depending on their specific features and use cases.

Figure 11 presents use case classification based on established networks.

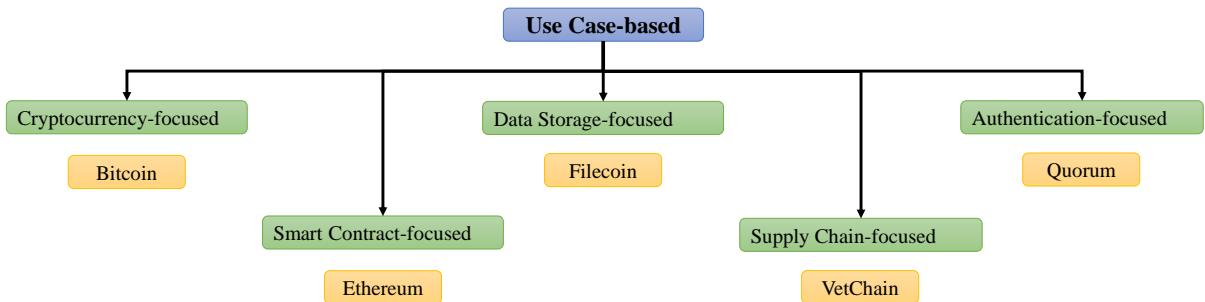


Fig. 11. Use-Based Classification.

Figure 12 illustrates a high level classification of DLT in general.

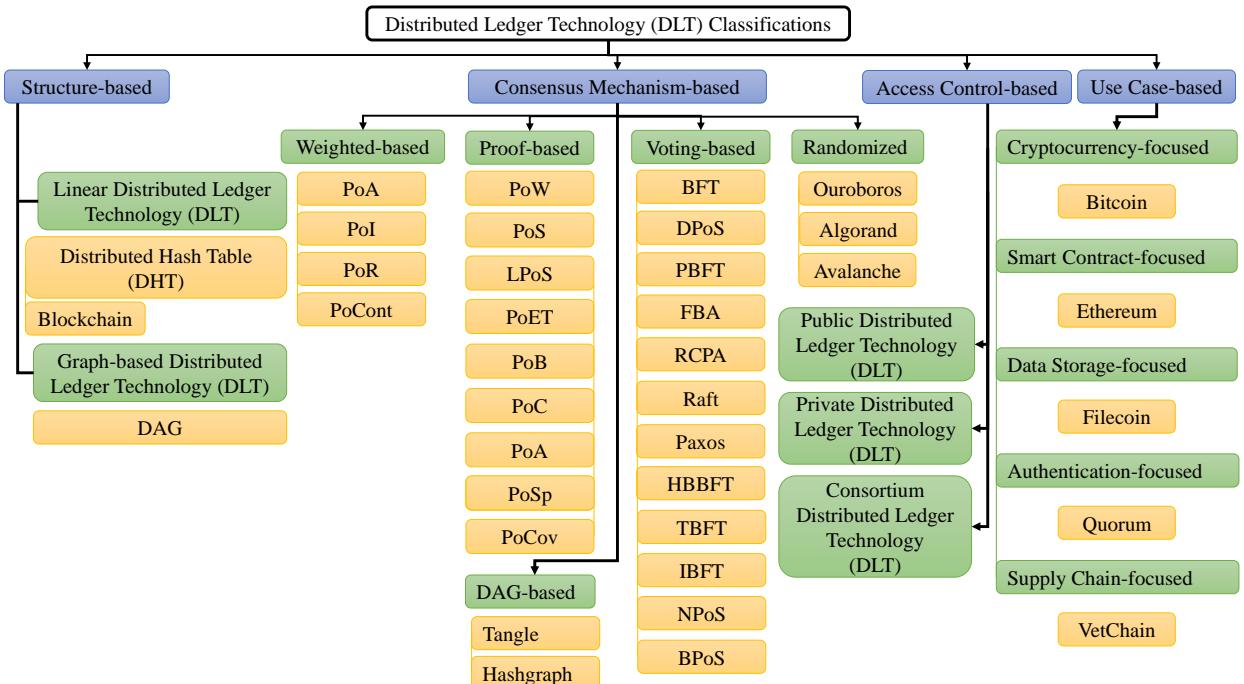


Fig. 12. DLT Classifications.

## V. General Overview for Consensus Algorithms

There are many consensus algorithms in use, with new ones being developed as the field of distributed ledger technologies evolves. While it is challenging to list all consensus algorithms in the market, some of the most common and notable ones are listed in Table III based on their primary and secondary categories.

### A. Proof of Work (PoW)

Proof of Work (PoW), is a consensus technique utilized by numerous blockchain networks, including Bitcoin and Ethereum 1.0. The PoW algorithm enables nodes, known as miners, to compete in solving challenging cryptographic problems as shown in figure 13 in order to validate transactions, generate new blocks, and secure the blockchain network [41]. The primary goal of PoW is to discourage malicious conduct by performing energy- and resource-consuming computationally intensive activities [42].

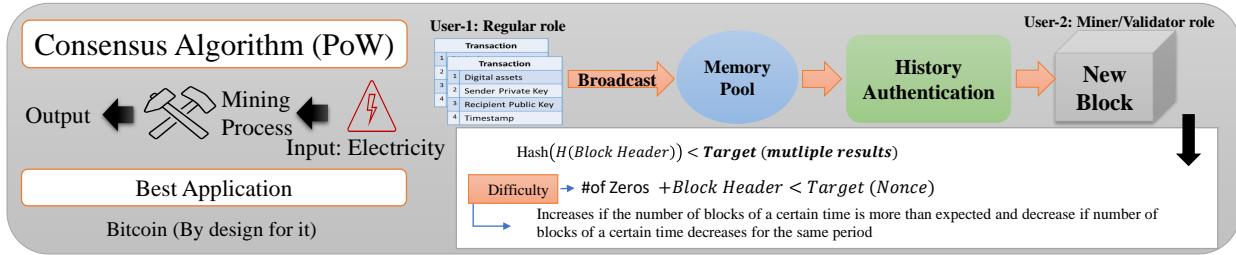


Fig. 13. PoW Operations .

The PoW consensus algorithm operates as follows:

- 1) Transaction Pool: Transactions are broadcast by users to the network. These transactions are then accumulated in a transaction pool, awaiting inclusion by miners in a subsequent block [43].
- 2) Block Candidate: A miner selects a group of unconfirmed transactions from the transaction pool, prioritizing those with the highest transaction fees. The miner then generates a block candidate, which consists of a block header containing essential data, such as the preceding block's hash, the Merkle tree root of the selected transactions, a timestamp, and a nonce [44].
- 3) Hash Function: Miners create a hash value for the block header using a cryptographic hash function, such as SHA-256 in the case of Bitcoin. A hash function accepts an input of arbitrary length and returns a fixed-length, seemingly random output (hash). Hash functions possess three fundamental properties: determinism, resistance to preimages, and collision resistance [45].
- 4) Proof of Work Puzzle: The PoW puzzle requires miners to identify a nonce, a number appended to the block header, such that the block header's resulting hash is less than or equal to a target value. The target value is determined by the network's current difficulty level, which regularly changes to maintain a constant block generation pace (e.g., approximately every 10 minutes for Bitcoin).
- 5) Mining: Miners continuously modify the nonce and recompute the hash of the block header until they find a valid solution that satisfies the desired condition. This procedure needs a considerable amount of energy and resources and is computationally intensive. The first miner to discover a correct solution may add the new block to the network.
- 6) Block Validation and Propagation: After discovering a valid solution, miners propagate the new block to the network. Other network nodes validate the block to ensure that the PoW solution is correct and that the transactions are genuine. If the block is genuine, it is added to the local copy of the blockchain, and nodes begin processing the next block using the hash of the newly added block as the reference in the new block header.
- 7) Block Reward: The miner who successfully adds a new block to the blockchain is rewarded with newly minted cryptocurrency (e.g., Bitcoin) and the total transaction fees from the included transactions. This payment acts as an incentive for miners to participate in the Proof-of-Work protocol and safeguard the network [46].

The most successful and widely recognized application of Proof of Work (PoW) consensus algorithm is Bitcoin because it has been designed for it. Rules embedded in the system such as longest chain and 10 minutes average per a block is to keep it sustainable and reliable until all 21 million coins are mined. The mechanism advantages and disadvantages are listed in table IV.

TABLE III  
CLASSIFICATION OF CONSENSUS ALGORITHMS

Algorithm	Primary Category	Secondary Category	Examples
Proof of Work (PoW)	Proof-based	NA	Bitcoin
Proof of Stake (PoS)	Proof-based	NA	Ethereum 2.0
Delegated Proof of Stake (DPoS)	Voting-based	Proof-based	EOS, Lisk, Tron
Leased Proof of Stake (LPoS)	Proof-based	Voting-based	Waves
Nominated Proof of Stake (NPoS)	Voting-based	NA	Polkadot
Bonded Proof of Stake (BPoS)	Voting-based	NA	Waves
Proof of Authority (PoA)	Weighted-based	NA	VeChain
Proof of Contribution (PoCon)	Weighted-based	Proof-based (PoW&PoS elements)	
Byzantine Fault Tolerance (BFT)	Voting-based	NA	
Practical BFT (PBFT)	Voting-based	NA	Hyperledger Fabric
Federated Byzantine Agreement (FBA)	Voting-based	NA	Stellar
Ripple Consensus Protocol Algorithm (RCPA)	Voting-based	NA	XRP Ledger
Raft Consensus Algorithm	Voting-based	NA	
Paxos	Voting-based	NA	
Honey Badger BFT	Voting-based	NA	
Avalanche Consensus	Voting-based	DAG-Based (PoW elements)	Avalanche platform
Proof of Elapsed Time (PoET)	Proof-based	NA	Intel's Sawtooth
Proof of Burn (PoB)	Proof-based	Voting-based	Slimcoin
Proof of Importance (PoI)	Weighted-based	NA	NEM
Proof of Capacity (PoC)	Proof-based	NA	Burstcoin
Proof of Activity (PoA)	Proof-based	NA	Decred, Espers
Proof of Space (PoSpace)	Proof-based	NA	Chia Network
Tendermint BFT	Voting-based	Proof-based (PoS elements)	Cosmos Network
Istanbul BFT	Voting-based	NA	Quorum
Proof of Reputation (PoR)	Weighted-based	NA	Idena
Proof of Coverage (PoC)	Proof-based	NA	Helium Network
Algorand	Randomized	Proof-Based (Pure PoS)	Algorand platform
Ouroboros	Randomized	Proof-based (PoS elements)	Cardano
Tangle	DAG-based	Proof-based (Lightweight PoW)	IOTA
Hashgraph	DAG-based	Voting-based (Original BFT)	Hedera

TABLE IV  
ADVANTAGES AND DISADVANTAGES OF POW.

Advantages	Disadvantages
Security	1. Energy consumption
Decentralization	2. Centralization risk
Sybil resistance	3. Hardware requirements
Proven track record	4. Inefficient resource allocation
Block appending timing	5. Limited throughput

## B. Lightweight PoW

Tangle consensus is unique to IOTA's Directed Acyclic Graph (DAG) data structure. This data structure lacks blocks and miners, unlike blockchains. Instead, transactions are interconnected in a web-like form. The consensus is a lightweight PoW that requires a participant to validate two previous transactions in order to add a new one [47]. The mechanism advantages and disadvantages are listed in table V.

TABLE V  
ADVANTAGES AND DISADVANTAGES OF TANGLE.

Advantages	Disadvantages
Scalability	Network maturity
No transaction fees	Coordinator reliance
Decentralization	Double-spending resistance concerns
Energy efficiency	Lower adoption
Adaptability	Complexity

## C. Proof of Stake (PoS) Family

Proof of Stake (PoS): A more energy-efficient alternative to PoW, PoS is designed to address some of the limitations of PoW, such as energy consumption and centralization risks. PoS relies on a participant's stake in the network, usually measured by the amount of cryptocurrency they hold or have locked up, to determine their chances of creating a new block and validating transactions. Many blockchain implementations, including Ethereum, have added smart contracts to the Bitcoin model. These smart contracts are self-executing and may be programmed to do a wide range of tasks, opening the door to new possibilities for automation and simplification of business operations [48]. The Proof of Stake consensus algorithm, depicted in figure 14, operates as follows:

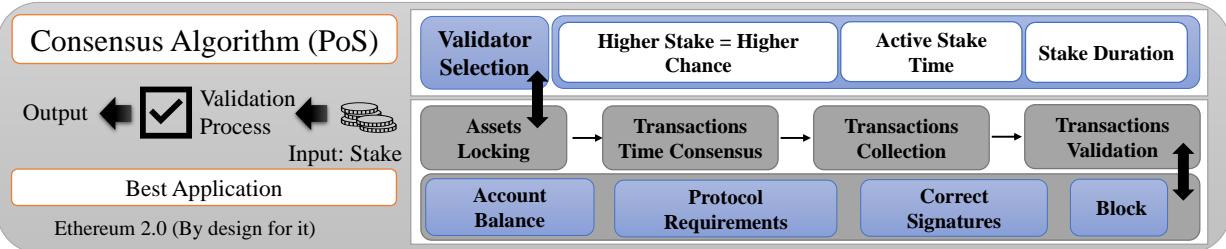


Fig. 14. PoS Operations.

- 1) Validators and Staking: Unlike PoW, where miners compete to solve cryptographic puzzles, PoS networks rely on validators to create new blocks and validate transactions. Validators are chosen based on the amount of cryptocurrency they hold and are willing to "stake" or lock up as collateral. Staking typically involves a validator depositing their cryptocurrency in a special wallet or smart contract, which is then locked for a predetermined period [48].
- 2) Selection Algorithm: Validators are chosen to create new blocks and validate transactions through a selection algorithm, which varies among PoS implementations. Common selection methods include randomization, coin age-based selection, and delegated proof of stake (DPoS), where token holders vote for validators [49].

- 3) Block Creation and Validation: The selected validator creates a new block containing a set of unconfirmed transactions and broadcasts it to the network. Other validators in the network check the block's validity, ensuring that the transactions are valid and the creating validator had the right to generate the block. If the block is valid, it is added to the blockchain, and the process continues with the selection of the next validator.
- 4) Block Rewards and Transaction Fees: In PoS networks, validators are usually rewarded with transaction fees from the transactions included in the block, rather than receiving newly minted cryptocurrency as in PoW systems. Some PoS implementations, however, may still offer block rewards in addition to transaction fees.
- 5) Slashing and Penalties: PoS networks often incorporate mechanisms to penalize validators who act maliciously or fail to validate transactions properly. This process, known as slashing, can result in the loss of a portion or all of the staked cryptocurrency, which serves as a strong deterrent against malicious behavior.
- 6) Security and Attacks: PoS networks are designed to be secure against attacks, such as 51% attacks and long-range attacks. The cost of acquiring a controlling stake in a PoS network is typically much higher than the potential rewards, making it economically irrational for an attacker to attempt to manipulate the network.
- 7) Energy Efficiency: PoS networks are generally more energy-efficient than PoW networks, as they do not require participants to expend large amounts of computational resources and energy to solve cryptographic puzzles. This makes PoS networks more environmentally friendly and cost-effective.

Various PoS implementations exist in the market, with Ethereum 2.0 (currently transitioning from PoW to PoS), Cardano, and Tezos being some notable examples. Each implementation may have its unique features and variations of the PoS algorithm, but the core principles remain the same. The mechanism advantages and disadvantages are listed in Table VI.

TABLE VI  
ADVANTAGES AND DISADVANTAGES OF POS.

Advantages	Disadvantages
1. Energy Efficiency	1. Initial Distribution
2. Decentralization	2. "Rich Get Richer" Effect
3. Security	3. Lack of Proven Track Record
4. Inflation Control	4. Complexity
5. Lower Entry Barrier	5. Centralization Risk

## 1) Delegated Proof of Stake (DPoS)

Delegated Proof of Stake (DPoS) is a consensus algorithm that builds upon the concepts of Proof of Stake (PoS), aiming to provide a more efficient, scalable, and democratic method of achieving consensus in a blockchain network. DPoS introduces a voting mechanism for selecting validators and focuses on a smaller number of elected delegates, which can result in faster transaction processing and increased accountability. DPoS was introduced by Daniel Larimer in 2014 as part of the BitShares blockchain platform [50]. The mechanism advantages and disadvantages are listed in Table VII.

TABLE VII  
ADVANTAGES AND DISADVANTAGES OF DPoS.

Advantages	Disadvantages
1. Scalability and Efficiency	1. Centralization Risk
2. Energy Efficiency	2. Voting Apathy and Voter Influence
3. Democracy and Accountability	3. Bribery and Corruption
4. Incentive to Act Honestly	4. Cartel Formation
5. Lower Entry Barrier	5. Complexity

## 2) Leased Proof of Stake (LPoS)

Leased Proof of Stake (LPoS) is a consensus mechanism that extends the concept of Proof of Stake (PoS) to allow users to lease their stakes to other network participants. This enables small stakeholders to participate in the staking process and earn rewards, even if they don't possess enough tokens to become validators themselves. LPoS was first implemented by the Waves Platform [51]. The mechanism advantages and disadvantages are listed in Table VIII.

TABLE VIII  
ADVANTAGES AND DISADVANTAGES OF LPOS.

<b>Advantages</b>	<b>Disadvantages</b>
1. Inclusivity	1. Complexity
2. Scalability and Efficiency	2. Trust in Validators
3. Security	3. Centralization Risk
4. Reward Distribution	4. Reduced Staking Control
5. Enhanced Decentralization	5. Liquidity Risk

### 3) Pure Proof-of-Stake (Algorand)

(PPoS) Pure Proof of Stake. This technique is different from traditional Proof of Stake (PoS) methods in that it ensures a high level of decentralization, security, and speed by choosing validators for block proposal and agreement in a random and secret way [52]. The mechanism advantages and disadvantages are listed in Table IX.

TABLE IX  
ADVANTAGES AND DISADVANTAGES OF PPOS.

<b>Advantages</b>	<b>Disadvantages</b>
Scalability	Wealth Concentration
Decentralization	Initial Distribution
Energy Efficiency	Network Maturity
Security	Potential for Centralization
Inclusivity	Barrier to Entry

### 4) Ouroboros

Ouroboros is a Proof of Stake (PoS) consensus mechanism. The protocol is made to keep decentralization while providing high levels of security, scalability, and energy economy. The procedure picks a committee of stakeholders at random for each slot, and the committee members decide together which block goes next in the chain. A random seed, which is changed at the end of each epoch, is used to choose the committee members [53]. The mechanism advantages and disadvantages are listed in Table X.

TABLE X  
ADVANTAGES AND DISADVANTAGES OF THE OUROBOROS CONSENSUS MECHANISM.

<b>Advantages</b>	<b>Disadvantages</b>
Energy Efficiency	Complexity
Security	Potential centralization risk
Decentralization	Initial bootstrapping
Scalability	Network connectivity
Incentivization	Adaptation

### 5) Nominated Proof of Stake (NPoS)

Nominated Proof of Stake (NPoS) is a variant of the Proof of Stake (PoS) consensus algorithm, with some unique features that differentiate it from other PoS-based mechanisms. It was introduced by the Polkadot network, a scalable and interoperable blockchain platform. NPoS is designed to provide a secure, scalable, and efficient consensus mechanism for blockchain networks. It is particularly suitable for platforms like Polkadot, which focus on interoperability and cross-chain communication [54]. The mechanism advantages and disadvantages are listed in Table XI.

### 6) Bonded Proof of Stake (BPos)

Bonded Proof of Stake (BPos) is a variation of the Proof of Stake (PoS) consensus algorithm, in which validators are required to lock up or bond a certain amount of their tokens as collateral. This bond acts as a security deposit, incentivizing validators to act honestly and discouraging malicious behavior. If a validator is found to be acting

TABLE XI  
ADVANTAGES AND DISADVANTAGES OF THE NPOS CONSENSUS MECHANISM.

<b>Advantages</b>	<b>Disadvantages</b>
Decentralization	Complexity
Security	Potential centralization of power
Scalability	Slashing risks
Inclusive participation	Inflation
Incentive alignment	Barrier to entry

maliciously, their bonded tokens can be confiscated or slashed as a penalty. By requiring validators to have a stake in the network, BPoS aims to ensure the network's security and trustworthiness [55]. The mechanism advantages and disadvantages are listed in Table XII.

TABLE XII  
ADVANTAGES AND DISADVANTAGES OF THE BPoS CONSENSUS MECHANISM.

<b>Advantages</b>	<b>Disadvantages</b>
Enhanced Security	Centralization Risk
Energy Efficiency	Barrier to Entry
Incentivized Participation	Slashing Penalties
Sybil Attack Resistance	Limited Token Circulation

Table XIII compares all Proof of Stake (PoS) variants in term of their operation and highlights the differences.

TABLE XIII  
COMPARISON OF POS FAMILY IN TERM OF OPERATIONS [49].

<b>Differences</b>	Participation and Inclusivity	Validator Selection	Reward Distribution	Network Decentralization
<b>PoS</b> [56]	Limited to major stakeholders	Based on stake and selection algorithm	Based on stake	Larger validator set
<b>DPoS</b> [57]	Token holders can vote for validators	Voting system to elect delegates	Elected delegates receive rewards	Smaller number of elected delegates
<b>LPoS</b> [51]	Users can lease their stakes to validators or staking pools	Similar to PoS, but includes leased stakes	Validators receive rewards based on total stake and distribute among leasing users	Similar to PoS, larger validator set
<b>PPoS</b>	All token holders can participate	Cryptographic sortition	Proportional to stake	High decentralization
<b>Ouroboros</b>	All token holders can participate	Based on stake and randomization	Proportional to stake	High decentralization
<b>NPoS</b>	Nominated validators	Nominated by token holders	Proportional to stake	Decentralization based on nominations
<b>BPoS</b>	Validators post bonds	Validators with bonded stakes	Proportional to bonded stake	Decentralization based on bonded stakes

#### D. Byzantine Fault Tolerance (BFT) Family

The original Byzantine Fault Tolerance (BFT) algorithm was introduced by Castro and Liskov in their 1999 paper "Practical Byzantine Fault Tolerance". The algorithm was designed to provide a robust and secure consensus mechanism for distributed systems in the presence of malicious or faulty nodes [58]. The original BFT algorithm operates in three phases: request, pre-prepare, and commit. The steps of each phase are as follows [59], [60]:

- 1) Request Phase: A client sends a request to the primary node, which is responsible for proposing a set of transactions. The request includes a sequence number and a unique client ID.

- 2) Pre-Prepare Phase: The primary node receives the request from the client and assigns a sequence number. It then creates a message that includes the sequence number, the proposed transaction set, and a digest of the previous message. The primary node sends this message to the backup nodes for validation.
- 3) Prepare Phase: Upon receiving the message from the primary node, the backup nodes validate the message by checking the sequence number, the digest, and the proposed transaction set. If the message is valid, the backup nodes send a “prepare” message to all other nodes in the network, indicating their agreement with the proposed transaction set.
- 4) Commit Phase: Once the backup nodes have reached a two-thirds majority agreement, they send a “commit” message to all other nodes in the network, indicating that the proposed transaction set has been approved. At this point, the transaction set is added to the ledger and can be queried by other nodes in the network.

One of the key features of the original BFT algorithm is that it requires a two-thirds majority of the backup nodes to agree on a proposed transaction set before it can be approved. This ensures that the network remains secure even if up to one-third of the nodes are faulty or malicious. Another important aspect of the original BFT algorithm is fault detection and recovery. BFT algorithms include mechanisms for detecting Byzantine faults, such as timeouts or replica divergence detection. If a fault is detected, the system can take corrective action, such as removing the faulty node or resynchronizing the state of the system.

The original BFT algorithm is a robust and secure consensus mechanism that enables distributed systems to reach agreement on a set of transactions, even in the presence of malicious or faulty nodes. Its technical operations include client request, pre-prepare, prepare, and commit phases, as well as fault detection and recovery mechanisms [61] [60]. Its advantages and disadvantages are listed in Table XIV.

TABLE XIV  
ADVANTAGES & DISADVANTAGES OF BFT ALGORITHM [62].

Advantages [58]	Disadvantages [58].
Robustness	Complexity
Security	Resource-intensive
Speed	Limited scalability
Fault tolerance	Single point of failure (primary node vulnerability)

## 1) Practical Byzantine Fault Tolerance (PBFT)

Practical Byzantine Fault Tolerance (PBFT) is a consensus algorithm that extends the original Byzantine Fault Tolerance (BFT) algorithm to improve scalability and reduce communication overhead. PBFT is designed to be used in permissioned blockchain networks, where the network participants are known and trusted [63]. The advantages and disadvantages are listed in Table XV. Figure 15 illustrates the operation of PBFT as a major variant of BFT and the most common one among the family.

TABLE XV  
ADVANTAGES & DISADVANTAGES OF PBFT ALGORITHM.

Advantages	Disadvantages
High Security and Reliability	Implementation Complexity
Fast Processing	Resource Intensive
Scalability	Limited Decentralization
Well-suited for Permissioned Networks	Centralization

## 2) Federated Byzantine Agreement (FBA)

Federated Byzantine Agreement (FBA) is variation of BFT. It involves a network of nodes that form quorum slices to reach consensus. Stellar uses a version of FBA called the Stellar Consensus Protocol (SCP). FBA is a consensus mechanism that operates differently from traditional Byzantine Fault Tolerance (BFT) algorithms. Instead of requiring every node to communicate with every other node, FBA allows nodes to choose their quorum slices, leading to a more decentralized and efficient consensus process [64]. The advantages and disadvantages are listed in Table XVI.

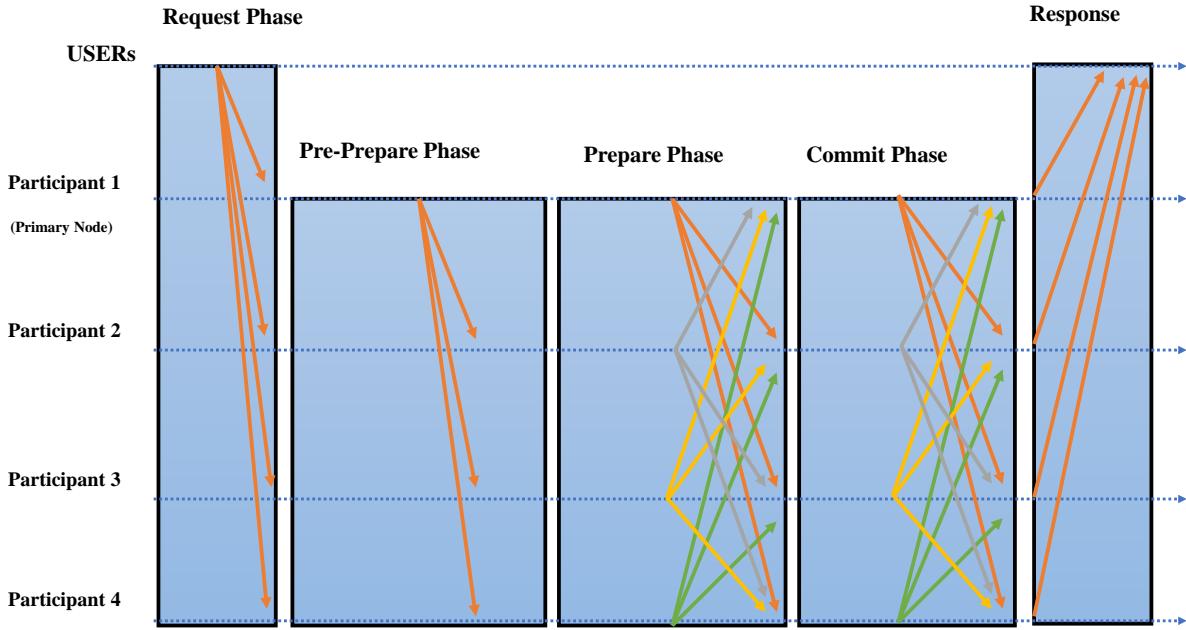


Fig. 15. PBFT Operations.

TABLE XVI  
ADVANTAGES AND DISADVANTAGES OF FBA ALGORITHM.

Advantages	Disadvantages
Scalability: Offers better scalability due to reduced communication overhead.	Complexity: Can be complex to implement and understand.
Decentralization: Promotes decentralization by allowing nodes to choose their own quorum slices.	Potential Network Splits: Poorly chosen quorum slices can lead to network splits.
Asynchronous Operation: Resilient to network latency and functions efficiently in an asynchronous environment.	Slower Finality: May take longer to reach consensus depending on the network configuration and quorum slices.
Flexible Trust: Allows nodes to determine their own trust relationships for a more robust network.	Vulnerability to Sybil Attacks: More vulnerable if malicious nodes can create enough fake nodes to influence quorum slices.

### 3) Istanbul BFT

Istanbul Byzantine Fault Tolerance (IBFT) is a consensus algorithm designed for blockchain networks with permissioned access. Practical Byzantine Fault Tolerance (PBFT) is a variant that provides high security, fault tolerance, and rapid transaction processing. Nodes in IBFT use a leader-based methodology to propose and validate blocks. As long as fewer than one-third of the nodes are flawed or malicious, the algorithm assures the system's functionality. IBFT is ideally adapted for enterprise-level applications in which network participants are known and trusted [65] [66]. The advantages and disadvantages are listed in Table XVII.

### 4) Honey Badger BFT

Honey Badger Byzantine Fault Tolerance (HBBFT) is an asynchronous consensus technique that tolerates Byzantine errors without compromising efficiency, durability, or security. For distributed systems with unreliable networks, HBBFT can handle arbitrary message delays.

TABLE XVII  
ADVANTAGES AND DISADVANTAGES OF IBFT ALGORITHM.

<b>Advantages</b>	<b>Disadvantages</b>
High Security and Reliability	Implementation Complexity
Fast Transaction Processing	Resource Intensive
Scalability	Limited Decentralization
Energy Efficiency	Centralization Risk
Finality	

Using threshold cryptography, HBBFT nodes generate batches of encrypted transactions. Each node contributes transactions and decrypts the bulk without exposing individual transactions. Decryption submits the batch to the blockchain.

Honey Badger BFT's strengths are its ability to withstand up to one-third of nodes being defective or malicious, its tolerance to network latency, and its excellent security guarantees. It is used to build distributed systems and blockchain networks like Nym [67]. The advantages and disadvantages are listed in Table XVIII.

TABLE XVIII  
ADVANTAGES AND DISADVANTAGES OF HBBFT ALGORITHM.

<b>Advantages</b>	<b>Disadvantages</b>
Asynchronous communication	Complexity
Robustness	Resource-intensive
Security	Limited decentralization
Adaptability	Communication overhead
High throughput	Less mature

## 5) Ripple Consensus Protocol Algorithm (RCPA)

Ripple Consensus Protocol Algorithm (RCPA) is the consensus mechanism used by the Ripple network (XRP Ledger) to validate transactions and maintain its distributed ledger. Ripple is a blockchain-based platform designed primarily for cross-border payments and real-time gross settlement. RCPA is a unique consensus mechanism that differs from Proof of Work (PoW), Proof of Stake (PoS), and other commonly known consensus algorithms. RCPA is a Byzantine fault-tolerant (BFT) consensus algorithm that uses a voting process among a group of validator nodes to agree on the state of the ledger [68]. The advantages and disadvantages are listed in Table XIX.

TABLE XIX  
ADVANTAGES & DISADVANTAGES OF RCPA ALGORITHM.

<b>Advantages</b>	<b>Disadvantages</b>
Fast transaction confirmations	Centralization
Energy efficiency	Trust in validators
Scalability	Limited decentralization benefits
Resistance to certain attacks	Control over validators
Lower transaction fees	

## E. Ignite (Tendermint)

Tendermint is a Byzantine Fault-Tolerant (BFT) consensus algorithm and a blockchain engine that lets developers build decentralized applications in a safe and efficient manner. Tendermint uses a BFT mechanism called Tendermint Core that can be used in both private and public blockchain networks [69]. The advantages and disadvantages are listed in Table XX.

## F. Hashgraph

Hashgraph is a distributed ledger technology (DLT) that uses a consensus method called “Swirls Consensus Algorithm.” It is an alternative to standard blockchain networks that promises to be faster, safer, and more fair.

TABLE XX  
ADVANTAGES AND DISADVANTAGES OF TENDERMINT ALGORITHM.

<b>Advantages</b>	<b>Disadvantages</b>
Fast finality	Centralization concerns
Scalability	Vulnerability to long-range attacks
Energy efficiency	Validator collusion
Byzantine Fault Tolerance	Initial validator setup
Easier upgrades and governance	Staking requirements

Hashgraph uses a data format called a Directed Acyclic Graph (DAG), in which transactions are arranged in a graph instead of in a chain of blocks. Swirls Consensus Algorithm is an Asynchronous Byzantine Fault Tolerance (aBFT) [70]. The advantages and disadvantages are listed in Table XXI.

TABLE XXI  
ADVANTAGES AND DISADVANTAGES OF HASHGRAPH.

<b>Advantages</b>	<b>Disadvantages</b>
High throughput	Centralization (patented technology)
Fairness	Licensing model
Asynchronous Byzantine Fault Tolerance (aBFT)	Complexity
No mining (energy-efficient)	Limited adoption

## G. Avalanche

Avalanche is a consensus system that aims to provide high throughput, low latency, and strong security. It is made for different kinds of decentralized applications and works best on open networks that don't require permission. The Avalanche consensus is a new way to reach a decision. It is a leaderless, metastable, Byzantine Fault Tolerant protocol [71]. The advantages and disadvantages are listed in Table XXII.

TABLE XXII  
ADVANTAGES & DISADVANTAGES OF AVALANCHE.

<b>Advantages</b>	<b>Disadvantages</b>
High Scalability	Relatively New Technology
Decentralization	Unclear Long-term Stability
Quick Finality	Security Dependence on Network Size
Energy Efficiency	Complexity in Protocol Design
Adaptive Consensus Mechanism	Adoption Challenges

Table XXIII compares BFT variants based on their operation and network complexity.

## H. Crash Fault Tolerance (CFT) Family

Crash Fault Tolerance (CFT) is a group of consensus algorithms that can handle faults in distributed systems that are not Byzantine. Byzantine Fault Tolerance (BFT), on the other hand, deals with random faults, such as malicious behavior. CFT focuses on handling crashes, failures, and nodes that stop responding. The CFT family is made up of consensus algorithms that keep the system running and consistent even if some nodes crash or stop working [72].

Algorithm operations in CFT involve the following steps:

- 1) Communication: Nodes in the network communicate with each other by exchanging messages. They rely on reliable and timely message delivery to coordinate their actions and reach consensus.
- 2) Agreement: Nodes must agree on a specific value or state proposed by a designated leader or a group of nodes. This can be achieved through voting, where nodes cast their votes for a proposal, or through other coordination mechanisms, like rounds of message exchanges.
- 3) Failure detection: Nodes monitor each other's behavior to detect failures. If a node doesn't receive a message within a specified timeframe or fails to respond to queries, it may be assumed to be unresponsive or crashed.

TABLE XXIII  
COMPARISON OF BFT FAMILY IN TERM OF OPERATIONS.

Differences	Voting	Network	Users	Complexity	Threshold
<b>BFT</b>	Rounds of voting	Sync. network	All nodes	$O(n^2)$	Up to 1/3
<b>PBFT</b>	Voting phases	Sync. network	All nodes	$O(n^2)$	Up to 1/3
<b>BFA</b>	Quorum slices and intersecting quorums	Partially sync. network	Quorum slices of nodes	$O(n)$	Configurable
<b>IBFT</b>	Round-robin leader selection	Sync. network	All nodes	$O(n^2)$	Up to 1/3
<b>HBBFT</b>	Parallel proposals and reconstruction	Async. network	All nodes	$O(n^2)$	Up to 1/3
<b>RCPA</b>	Trusted validators propose and vote on transactions	Sync. or partially sync. network	Trusted validators	$O(n)$	Configurable
<b>Ignite</b>	Propose and vote on blocks	Partially sync. network	Validators with voting power	$O(n^2)$	Up to 1/3
<b>Hashg.</b>	Gossip-based voting	Async. network	All nodes	$O(n * \log(n))$	Up to 1/3
<b>Aval.</b>	Chained consensus with multiple subnets	Partially sync. network	Validators with voting power	$O(n)$	Up to 1/2

- 4) Recovery: Once a failure is detected, the remaining nodes work together to recover from the crash fault. This may involve electing a new leader, redistributing the workload, or reconfiguring the system to continue operation without the faulty node.
- 5) Consistency: Despite the presence of crash faults, the non-faulty nodes must ensure that the system maintains consistency. This means that all nodes must eventually agree on a consistent state or value, even if some nodes have crashed or become unresponsive.

CFT consensus algorithms are designed to handle non-malicious failures in distributed systems. However, they are not equipped to deal with Byzantine faults, where nodes can behave arbitrarily or maliciously. For systems that need to withstand Byzantine faults, Byzantine Fault Tolerance (BFT) algorithms are used instead. Table XXIV lists advantages and disadvantages of the algorithm.

TABLE XXIV  
ADVANTAGES AND DISADVANTAGES OF CFT ALGORITHM.

Advantages	Disadvantages
Simplicity	Limited fault tolerance
Efficiency	Trust assumptions
Robustness	Vulnerability to attacks
Faster consensus	Dependence on timely communication

## 1) Paxos

A variant of CFT and an algorithm for getting all the nodes in a distributed system to agree on a single value or operation, even if some of the nodes are broken. It is especially useful for fault-tolerant distributed systems where nodes can fail or become unreachable [73] [74]. Table XXV lists advantages and disadvantages of the algorithm.

## 2) RAFT

RAFT is a distributed consensus algorithm that is made to be easy to understand and use while ensuring fault tolerance, safety, and liveness. It can be used instead of the Paxos algorithm, which is more complicated [75] [76]. Table XXVI lists advantages and disadvantages of the algorithm.

In Table XXVII the differences are represented in term of operations between CFT variants.

TABLE XXV  
ADVANTAGES AND DISADVANTAGES OF PAXOS ALGORITHM.

<b>Advantages</b>	<b>Disadvantages</b>
Fault Tolerance	Complexity
Safety	Liveness Issues
Asynchronous Communication	Communication Overhead
Adaptable	Dependency on a Leader

TABLE XXVI  
ADVANTAGES AND DISADVANTAGES OF RAFT ALGORITHM.

<b>Advantages</b>	<b>Disadvantages</b>
Understandability	Performance
Fault Tolerance	Scalability
Strong Consistency	Partial Network Partitions
Leader Election	Complexity
Dynamic Membership Changes	

## I. Proof of Elapsed Time (PoET)

Proof of Elapsed Time (PoET) is a consensus algorithm initially developed by Intel for permissioned blockchain systems. It is designed to provide a fair, energy-efficient, and highly scalable alternative to Proof of Work (PoW) and other consensus mechanisms. PoET leverages Intel's Software Guard Extensions (SGX) technology, a set of hardware extensions providing secure enclaves to protect sensitive data and code execution from external access or tampering [77].

How the Proof of Elapsed Time Algorithm operates:

- 1) Trusted Execution Environment (TEE): PoET relies on a Trusted Execution Environment (TEE) provided by Intel SGX, which enables the execution of trusted code in a secure enclave, ensuring the integrity and confidentiality of the code and data.
- 2) Wait Time Lottery: PoET operates as a “wait time lottery,” where participating nodes (validators) generate a random wait time and go to sleep for that duration. The validator with the shortest wait time wakes up first, creates a new block, and broadcasts it to the network. Other validators verify the block and add it to the blockchain if it is valid. The process then repeats for the next block.
- 3) Wait Time Calculation: Each validator generates its random wait time within the secure enclave using the TEE. This wait time calculation uses the previous block’s data, the validator’s local information, and a target wait time set by the network. The resulting wait time is cryptographically signed by the enclave to ensure its authenticity and prevent tampering.

TABLE XXVII  
COMPARISON OF CFT FAMILY IN TERM OF OPERATIONS.

<b>Differences</b>	<b>Goal</b>	<b>Model</b>	<b>Roles</b>	<b>Protocol</b>	<b>Recovery</b>
CFT	Ensure system reliability despite crash failures	Assumes crash failures only	No specific roles	Varies based on specific implementation	Relies on replication and backups
Paxos	Ensure consistency and agreement in distributed systems	Assumes crash failures only	Proposers, Acceptors, Learners	Three phases: Prepare, Propose, Learn	Learners apply committed decisions
RAFT	Ensure consistency and agreement in distributed systems	Assumes crash failures only	Leader, Followers, Candidates	Three phases: Leader Election, Log Replication, Safety	Nodes store and apply committed entries

- 4) Winning Validator: When a validator's wait time elapses, it wakes up and creates a new block proposal. The block proposal includes the validator's wait time, the secure enclave's signature, and the transactions to be included in the block. The validator then broadcasts its block proposal to the network.
- 5) Block Verification: Other validators in the network verify the block proposal by checking the following: The block proposal's wait time is less than or equal to the target wait time. The secure enclave's signature is valid, confirming that the wait time was generated within a TEE. The transactions in the block are valid. If the block proposal passes the verification, validators add the block to their local copy of the blockchain.

The primary advantages of PoET are its energy efficiency, as it does not require the computationally intensive mining process of PoW, and its fairness, as each participating node has an equal chance of creating a block. However, PoET's reliance on Intel's SGX technology raises concerns about centralization, as it requires specific hardware and trust in Intel as the technology provider. Additionally, potential vulnerabilities in SGX could compromise the security and integrity of the PoET consensus mechanism [78]. Table XXVIII lists advantages and disadvantages of the algorithm.

TABLE XXVIII  
ADVANTAGES AND DISADVANTAGES OF POET.

Advantages	Disadvantages
Energy efficiency	Dependence on Intel SGX
Fairness	Trust in Intel
Scalability	SGX Vulnerabilities
Security	Permissioned networks

## J. Proof of Burn (PoB)

Proof of Burn (PoB) is an alternative consensus algorithm that aims to address some of the drawbacks of Proof of Work (PoW), such as energy inefficiency and the centralization of mining power. PoB requires participants to “burn” or permanently destroy a certain amount of cryptocurrency to obtain the right to create new blocks and earn rewards. This process simulates the consumption of resources, like PoW, but without the significant energy expenditure associated with mining [79].

How the Proof of Burn Algorithm operates:

- 1) Coin Burning: In PoB, participants (validators or miners) destroy a certain amount of cryptocurrency, usually by sending it to an unspendable address (known as an “eater address”). This process is called “burning” the coins, as they are removed from circulation and become irretrievable.
- 2) Burned Coins as Proof: The act of burning coins serves as a proof of the participant's commitment to the network. The more coins a participant burns, the higher their chances of being selected to create a new block. This simulates the resource consumption of PoW but without the energy-intensive mining process.
- 3) Block Creation: The protocol periodically selects a participant to create a new block based on the amount of cryptocurrency they have burned. This selection can be done using various algorithms, such as a weighted random approach where the probability of being chosen is proportional to the amount of cryptocurrency burned.
- 4) Block Rewards: The selected participant creates a new block and receives a block reward, typically in the form of newly minted cryptocurrency. This incentivizes participants to burn coins, as they have the potential to earn more coins in the long run by creating new blocks.
- 5) Burn-and-Mint Equilibrium: Over time, an equilibrium is reached between the amount of cryptocurrency burned and the amount minted through block rewards. This equilibrium helps regulate the rate of coin burning and ensures that the overall cryptocurrency supply remains stable.

Proof of Burn is an alternative consensus mechanism that addresses some of the challenges associated with Proof of Work, such as energy inefficiency and centralization. By requiring participants to burn cryptocurrency to gain the right to create new blocks, PoB offers a more energy-efficient approach to achieving consensus while still simulating resource consumption. However, the permanent destruction of cryptocurrency and potential wealth concentration issues are significant concerns that may limit its adoption in certain applications [79]. Table XXIX lists advantages and disadvantages of the algorithm.

TABLE XXIX  
ADVANTAGES AND DISADVANTAGES OF PoB [80], [81], [82].

Advantages	Disadvantages
Energy Efficiency	Coin Destruction
Reduced Centralization	Wealth Concentration
ASIC Resistance	Limited Adoption
Security	Barrier to Entry

## K. Proof of Coverage (PoC)

Proof of Coverage (PoC) is a consensus algorithm used predominantly in wireless networks, such as the Helium blockchain, to validate network participant coverage. The algorithm incentivizes participants to deploy wireless coverage devices known as hotspots that provide connectivity for Internet of Things devices. These locations are rewarded for demonstrating their coverage area and service quality [83]. The Proof of Capacity Algorithm operates as follows:

- 1) Challenge Creation: The network or hotspot generates a challenge consisting of encrypted data that must be transmitted and received by hotspots.
- 2) Beacon Transmission: The challenged hotspot transmits a beacon comprising the encrypted data as evidence that it has transmitted the signal. As evidence, the transaction is recorded on the blockchain.
- 3) Witnessing: Neighboring locations within the coverage area receive and act as witnesses for the beacon. They affirm receipt of the beacon and submit their validation to the blockchain to validate the transmission.
- 4) Reward Distribution: Hotspots that participate in the challenge-response process by transmitting beacons and witnessing transmissions are rewarded with the native cryptocurrency (e.g., Helium tokens, HNT).

Its applicability is limited to specific use cases and relies on the assumption that participants act honestly in providing accurate information about their coverage area and service quality. Table XXX lists advantages and disadvantages of the algorithm.

TABLE XXX  
ADVANTAGES AND DISADVANTAGES OF PoC.

Advantages	Disadvantages
Incentivized Network Expansion	Limited Applicability
Energy Efficiency	Trust in Participants
Decentralization	Geographical Limitations
Security	Sybil Attacks
Cost-Effectiveness	

## L. Proof of Capacity (PoC)

Proof of Capacity (PoC) is a consensus algorithm used in some blockchain networks as an alternative to the energy-intensive Proof of Work (PoW) mechanism. Also known as Proof of Space, PoC relies on the available storage capacity of network participants to achieve consensus and maintain the network's security [84], [85]. The Proof of Capacity Algorithm operates as follows:

- 1) In PoC, participants, called “farmers,” allocate a portion of their storage capacity to store a large dataset called “plot” files. These plot files contain unique solutions to cryptographic problems, similar to nonces in the PoW mechanism. When a new block needs to be added to the blockchain, the network initiates a competition among the farmers. The farmers scan their plot files to find the best solution for the cryptographic problem presented in the new block.
- 2) The winning farmer is the one who finds the solution with the shortest deadline, which is a specific value associated with the solution. The deadline represents the time it takes for the farmer to mine the block. The shorter the deadline, the faster the farmer can mine the block. The winning farmer then has the right to add the block to the blockchain and receive the mining reward.
- 3) To participate in PoC, farmers must perform a one-time process called “plotting.” During plotting, the farmer generates the plot files by computing all possible solutions for cryptographic problems and storing them on

their allocated storage. The plotting process is computationally intensive but only needs to be done once, after which the farmer can participate in the PoC network with minimal resource usage.

- 4) Energy Efficiency: PoC is more energy-efficient than PoW, as it requires less computational power for mining. Instead of constantly solving complex problems using a large amount of processing power, PoC requires only storage space and low-power processing to read from the storage.
- 5) Fairness and Decentralization: PoC provides a more level playing field for participants, as it does not require specialized mining hardware like PoW. This can result in increased decentralization, as more participants can afford to join the network, reducing the influence of large mining pools.
- 6) Lower Barrier to Entry: As PoC relies on storage capacity rather than computational power, the barrier to entry is lower for participants. Many individuals already own storage devices with unused capacity, making it easier for them to join a PoC-based network.
- 7) Hardware Longevity: PoC-based mining is less demanding on hardware compared to PoW mining, which can lead to longer-lasting hardware and reduced electronic waste [86].

Table XXXI lists advantages and disadvantages of the algorithm.

TABLE XXXI  
ADVANTAGES AND DISADVANTAGES OF POC [84], [86], [87].

<b>Advantages</b>	<b>Disadvantages</b>
Energy Efficiency	Storage Centralization
Fairness and Decentralization	Data Storage Inefficiency
Lower Barrier to Entry	Vulnerability to Sybil Attacks
Hardware Longevity	Slower Adoption

## M. Proof of Importance (PoI)

Proof of Importance (PoI) is a consensus mechanism used in certain blockchain networks, including NEM, to determine which participants have the authority to establish new blocks and validate transactions. PoI is designed to recompense active network participants, encourage more equitable reward distribution, and discourage coin hoarding [88].

The PoI algorithm operation is as follows:

- 1) Account balance evaluation: The PoI algorithm considers each participant's account balance when determining their significance. A participant's importance score is greater if they have a greater interest in the network.
- 2) Transaction Activity evaluation: The PoI algorithm investigates each participant's transaction history, with more active participants receiving higher importance scores. This rewards users who initiate transactions and actively contribute to the network.
- 3) Network Topology Consideration: The PoI algorithm takes into account the significance of other network participants with whom the user has conducted business. This encourages users to conduct business with other influential and active users, thereby enhancing the network.
- 4) Importance: Calculation of importance score The PoI algorithm incorporates the aforementioned factors to determine an importance score for each participant. This score represents the participant's contribution to the network and is used to determine the participant's likelihood of being selected to create new blocks and validate transactions.
- 5) Block validation: Participants with higher importance scores are more likely to be selected to create new blocks and validate transactions. This selection process promotes active network participation and a more decentralized, equitable distribution of rewards.

Table XXXII lists advantages and disadvantages of the algorithm.

## N. Proof of Reputation (PoR)

Proof of Reputation (PoR) is a consensus algorithm used in some blockchain networks to establish trust and determine who can produce new blocks and validate transactions. By rewarding participants with positive reputations, Proof of Reputation seeks to encourage network activity, security, and equity [89].

The PoR algorithm operation is the following:

TABLE XXXII  
ADVANTAGES AND DISADVANTAGES OF POI.

<b>Advantages</b>	<b>Disadvantages</b>
Incentivizes active nodes	Complexity
Fairness	Potential for manipulation
Decentralization	Limited adoption
Encourages network growth	Initial importance assignment

- 1) Reputation assessment: The PoR algorithm evaluates each participant's reputation based on their past actions, compliance with network norms, and contribution to the network as a whole. This may include transaction history, up-time, and other metrics defined by the particular implementation.
- 2) Calculation of reputation scores: The algorithm computes a reputation score for each participant that reflects their credibility and prominence within the network. The precise method for calculating reputation scores may differ between implementations, but typically involves combining multiple factors into a holistic evaluation.
- 3) Reputation: Frequently, PoR systems employ a weighting mechanism that assigns greater weights to individuals with higher reputation scores. This weight affects the likelihood that a participant will be selected to generate new blocks and validate transactions.
- 4) Block validation: Participants with higher reputation scores (and, therefore, higher weights) have a greater chance of being selected to create new blocks and validate transactions. This selection procedure encourages network participants to maintain a positive reputation and promotes a more decentralized, secure, and equitable distribution of rewards.

Taking these pros and cons into account, PoR can be a good way to reach a consensus in apps that value trust, security, and decentralization. But in some situations, its complexity, ability to be manipulated, and the fact that image is hard to define may be problems [89]. Table XXXIII lists advantages and disadvantages of the algorithm.

TABLE XXXIII  
ADVANTAGES AND DISADVANTAGES OF POR.

<b>Advantages</b>	<b>Disadvantages</b>
Incentivizes good behavior	Complexity
Decentralization	Potential for manipulation
Security	Subjectivity
Adaptability	Initial reputation assignment

## O. Proof of Contribution (PoC)

Blockchain networks use Proof of Contribution (PoC) to reward network contributors. PoC rewards network validators or nodes that provide useful resources or services. Storage, processing power, and data transport are examples. PoC supports network engagement and decentralization by compensating contributors [90].

Operations of the PoC algorithm:

- 1) Contribution tracking: The algorithm keeps track of what each person has contributed. Contributions can be things like computing power, storage space, or network bandwidth, or they can be acts like validating transactions, taking part in governance, or providing other services.
- 2) Contribution scoring: The algorithm gives each person a score based on what they have contributed. This number shows how much the member has contributed to the network as a whole. It can be calculated using a weighted formula that looks at various types of contributions and how important they are to the network as a whole.
- 3) Selection: Choosing who gets rewards or tasks: The algorithm uses the input scores to choose who gets rewards or tasks in the network. For example, people with higher scores may be more likely to be chosen as validators or to get a share of newly created tokens.
- 4) Reward distribution: Once the players have been chosen, the algorithm divides the rewards based on how well they contributed. This can happen through the creation of new tokens or through transaction fees.
- 5) Updates and Adjustment: Updates and changes are always being made. The program is constantly updating the contribution scores and changing the way rewards are given based on how people are contributing. This makes sure that the network stays busy and gives participants a reason to stay involved.

Table XXXIV lists advantages and disadvantages of the algorithm.

TABLE XXXIV  
ADVANTAGES AND DISADVANTAGES OF PROOF OF CONT.

Advantages	Disadvantages
Incentivizes active participation	Complexity
Decentralization	Potential for gaming the system
Fair distribution of rewards	Difficulty in quantifying contributions
Encourages diverse contributions	Scalability concerns
Dynamic and adaptable	Centralization risks

Due to the implementation similarity of PoR, PoI, and PoCont, Table XXXV compares them based on their operations.

TABLE XXXV  
COMPARISON OF POR, POI, AND POC IN TERM OF OPERATIONS.

Differences	Basis for consensus	Calculation method	Incentives	Sybil attack resistance	Target use case
<b>PoR</b>	Based on the reputation of participants	Reputation determined by participant's history of positive contributions and penalties for negative behavior	Higher reputation scores receive more significant rewards	Resistant due to the difficulty of creating multiple fake identities with high reputation scores	Platforms that require trust and positive behavior, such as social networks and content curation platforms
<b>PoI</b>	Based on the importance of participants	Importance calculated using factors like account balance, transaction activity, and interacting node importance	Higher importance scores receive more significant rewards	Resistant due to the difficulty of creating multiple fake identities with high importance scores	Networks that aim to reward active and influential participants, such as NEM's blockchain
<b>PoC</b>	Based on the contributions of participants	Contribution determined by evaluating the value or resources provided by the participant	Participants with more contributions receive more significant rewards	Resistant due to the difficulty of creating multiple fake identities with high contribution scores	Networks that aim to reward participants based on their contributions, such as resource sharing or computational power

## P. Proof of Authority (PoA)

Proof of Authority (PoA) is a consensus algorithm used in permissioned blockchain networks, where a set of pre-approved, trusted nodes (known as authorities or validators) are responsible for validating transactions and creating new blocks. PoA emphasizes the identity and reputation of the validators as a means to ensure network security, rather than relying on computational power or stake as in Proof of Work (PoW) or Proof of Stake (PoS) systems [91].

How the Proof of Authority consensus algorithm operates:

- 1) Authority Selection: In a PoA-based blockchain, a limited number of validators are chosen based on their identity and reputation. These validators are usually known entities, such as organizations or individuals, with a high level of trustworthiness. The selection process can vary depending on the implementation, but it typically involves manual vetting, a governance mechanism, or a combination of both.
- 2) Block Creation: Validators take turns in a round-robin fashion to create new blocks. When a validator's turn comes, it collects and validates the pending transactions, then creates a new block that includes the

valid transactions, a reference to the previous block's hash, a timestamp, and the validator's digital signature. The digital signature serves as the validator's seal of approval, confirming that the block complies with the network's rules.

- 3) Block Validation: Once a new block is created, the other validators in the network verify the block by checking the transactions and ensuring that the digital signature is valid. If the majority of validators approve the block, it is added to the blockchain, and the transactions within it are considered confirmed.
- 4) Incentives and Penalties: Validators are typically rewarded for their work in validating and creating blocks. The rewards can be in the form of transaction fees or native cryptocurrency. On the other hand, if a validator behaves maliciously or fails to follow the network's rules, it may face penalties, such as being removed from the validator set or forfeiting a security deposit.
- 5) Governance: PoA networks often include governance mechanisms to manage the validator set, handle disputes, and make decisions regarding network upgrades or changes. Governance can involve voting by the validators themselves or other participants in the network, depending on the implementation [91].

Table XXXVI lists advantages and disadvantages of the algorithm.

TABLE XXXVI  
ADVANTAGES AND DISADVANTAGES OF POA.

<b>Advantages</b>	<b>Disadvantages</b>
Energy efficiency	Centralization
Scalability	Trust in authorities
Security	Limited decentralization benefits
Predictable block times	Vulnerability to regulatory pressure
Reduced risk of 51% attacks	Governance challenges

## Q. Proof of Activity (PoA)

Proof of Activity (PoA) is a hybrid consensus method that combines the best parts of Proof of Work (PoW) and Proof of Stake (PoS) to make a blockchain network that is more balanced and uses less energy. Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld came up with the idea in 2014 as an option to PoW and PoS [92].

The algorithm operations of PoA are as follows:

- 1) Mining: Like PoW, miners try solve a cryptographic puzzle. The first miner to figure out the puzzle makes a template block with the block header and the address of where to send the miner's prize.
- 2) Selection: Validators are picked based on how much they have invested in the network, according to the PoS protocol. The chance of being chosen as a validator goes greater when the stake higher.
- 3) Block signing: The validators who were chosen sign the sample block and add transactions to it. The deals are verified when each validator adds their signature to the block.
- 4) Validation threshold: A block is only considered acceptable if it has been signed by a certain number of validators, which is set by the network's consensus rules.
- 5) Block addition: The block is added to the blockchain when the minimum number of signatures are collected. Validators and miners both get a share of the block payment for what they do for the network.

Table XXXVII lists advantages and disadvantages of the algorithm.

TABLE XXXVII  
ADVANTAGES AND DISADVANTAGES OF POACT.

<b>Advantages</b>	<b>Disadvantages</b>
Energy Efficiency	Complexity
Security	Limited adoption
Decentralization	PoW drawbacks (energy consumption)
Incentivization	Governance challenges
Reduced risk of 51% attacks	Potential centralization

## R. Proof of Authentication (PoAh)

PoAh is a novel consensus algorithm that replaces PoW with a cryptographic authentication mechanism for resource-constrained devices, thereby rendering the blockchain application-specific. PoAh provides a decentralized security solution that is suitable for both private and permissioned blockchains, unlike other consensus algorithms [37] [38]. Table XXXVIII lists advantages and disadvantages of the algorithm.

TABLE XXXVIII  
ADVANTAGES AND DISADVANTAGES OF POAH.

Advantages	Disadvantages
More energy-efficient than PoW	Increased complexity compared to PoA
Handles large-scale IoT frameworks	Limited applicability to other blockchain networks
Faster block creation and transaction process	Only one factor (trust Score)
Suitable for resource-constrained IoT devices	Requires further research, development, and testing
Secure consensus through distributed trust	Centralization risk & potential for Sybil attacks

## S. Proof of PUF-Chain (PoP)

The focus of “PUFchain 2.0: Hardware-Assisted Robust Blockchain for Sustaining Simultaneous Device and Data Security in Smart Healthcare” is the development of a blockchain-based solution for enhancing device and data security in smart healthcare systems. PUFchain 2.0 is a hardware-assisted, secure blockchain that employs Physically Unclonable Functions (PUFs) to improve security [93] [94]. Table XXXIX lists advantages and disadvantages of the algorithm.

TABLE XXXIX  
ADVANTAGES AND DISADVANTAGES OF POP.

Advantages	Disadvantages
Enhanced security	Complexity
Device authentication	Cost
Data integrity and privacy	PUF limitations
Decentralization	Adoption challenges
Scalability	Potential for new attack vectors

## T. Proof of Block & Trade (PoBT)

IoT blockchain-specific lightweight proof of block and trade (PoBT) consensus algorithm. It is optimized for IoT systems, which are typically large-scale and heterogeneous networks of devices, in contrast to other consensus algorithms used in enterprise blockchains. PoBT attains scalability by simplifying the complex consensus-based security employed by conventional business blockchain algorithms [95]. It is a research based consensus algorithm. Table XL lists advantages and disadvantages of the algorithm.

TABLE XL  
ADVANTAGES AND DISADVANTAGES OF POBT.

Advantages	Disadvantages
Lightweight and scalable	Relies on business trust
Low energy consumption	Potential for trust manipulation
Low latency	Less suitable for non-business IoT applications

## VI. Our Proposed Methodology for Analysis of Consensus Algorithms

In this section, we demonstrate the approach followed to analyze application based multiple consensus algorithms from data collected in section V to generally evaluate the suitability of a particular consensus algorithm to a certain application.

### A. Multiple Attributes and Their Definitions

This section presents the attributes as shown in Table XLI and their levels to evaluate a certain consensus algorithm compared to the required consensus based on application conditions.

TABLE XLI  
ATTRIBUTES ASSOCIATED WITH CONSENSUS ALGORITHMS ANALYSIS.

Attribute	Definition
Hardware Requirements	Refer to node's computational resources, components, and infrastructure for network participation.
Pre-Trust Level	Refers to that each consensus mechanism requires a certain level of trust.
Tolerance Level	Refers to consensus mechanism's robustness and adaptability under adversary or system faults.
Overhead Computation	Refers to the need for additional resources to participate in the consensus and sustain the network.
Centralization Level	Refers to Control, decision-making, and resource concentration within a network.
Scalability Level	Refers to network's ability to handle more transactions and users while maintaining performance.
Latency Level	Refers to transaction's processing, confirmation, and ledger entry time.
Cost Level	Refers to the various expenses associated with operating, maintaining, and participating in a network.
Security Level	Refers to network's ability to withstand cyber-attacks, fraud, and manipulation.
Interoperability	Refers to adaptability to diverse communication protocols, data formats, and system architectures.
Complexity Level	Refers to the complexity associated with the implementation.

#### 1) Hardware Requirements

Hardware requirements vary depending on the consensus algorithm being used and can include aspects such as Computational Power, Storage Capacity, Energy Consumption, Network Connectivity, Specialized Hardware, and Stake or Investment. In this paper, it is divided to 5 levels: High, Moderate - High, Moderate, Low - Moderate, Low.

**Definitions:** High (H): Requires specialized hardware (ASICs/GPUs) for mining, high energy consumption. Moderate to High (M-H): Requires a computer or server with large storage capacity for plotting, lower energy consumption compared to PoW. Moderate (M): Requires a computer or server to run a validator node, staking an amount of cryptocurrency. Lower energy consumption. Low to moderate (L2M): Requires a computer, server, or IoT device to run a node, lower energy consumption compared to PoW. Low (L): CPS or IoT device to run a node.

#### 2) Pre-Trust Level

The level of trust required for each consensus algorithm varies depending on the network type, participants, and underlying assumptions. Generally, permissionless DLTs like PoW and PoS require no pre-established trust between participants, while permissioned DLTs using PBFT and its variants require higher levels of trust in the network's structure and participants.

**Definitions:** H: Trust is based on the identity and reputation of pre-selected validators. M2H: Trust is based on a list of trusted validator nodes chosen by each participant. M: Trust is based on the amount of cryptocurrency staked by validators; higher stake indicates higher trust. L2M: Trust is based on pre-existing trust in at least in early stages. L: Trust is based on computational power and solving mathematical puzzles; no pre-existing trust required.

#### 3) Tolerance Level

Tolerance level requirements can be categorized into the following aspects: Byzantine Fault Tolerance (BFT), Sybil Attack Resistance, Double Spending Resistance, Network Partition Tolerance, Censorship Resistance, Long-Range Attack Resistance, and Selfish Mining Resistance.

**Definitions:** H: Capable of tolerating up to a third of faulty or malicious nodes, highly resistant to attacks, and offering strong consistency and dependability. M2H: Tolerate a moderate number of malfunctioning nodes, achieve

a balance between security and performance, and are less susceptible to attacks than algorithms with low tolerance. M: Manage a limited number of defective nodes, achieve a balance between security and performance, and rely on network configuration and node truthfulness for security. L2M: Accept a small number of malfunctioning nodes, rely on trusted authorities or centralizing mechanisms, network may be compromised in certain attack scenarios. L: Handles very few or no defective nodes, is susceptible to a variety of attacks, prioritizes performance over security, and is not advised for high-value or mission-critical applications.

#### 4) Overhead Computation

Overhead computation can have a significant impact on the efficiency, scalability, and resource requirements of a DLT system. Different consensus algorithms have varying levels of overhead computation, which can influence the choice of consensus mechanism for a particular application. Overhead computation can be attributed to various aspects of the consensus process, such as Block validation, Block mining or forging, Network communication, Voting and agreement.

**Definitions:** H: require a significant amount of computational resources, power, or effort from nodes to participate in the consensus process. M2H: A slightly lower resource requirement than high overhead computation algorithms but still demand considerable computational resources from nodes. M:Require a fair amount of computational resources but are less resource-intensive compared to high overhead computation algorithms. L2M: Lower computational resource requirements compared to moderate overhead computation algorithms, but they still demand some level of computational effort from nodes. L: Requires minimal computational resources from nodes to participate in the consensus process.

#### 5) Centralization Level

A high level of centralization indicates that a small number of nodes or entities exert significant influence over the network, while a low level of centralization suggests a more evenly distributed control among participants. Centralization level is an essential factor in the design and evaluation of consensus algorithms, as it impacts various aspects of a DLT system, such as, Security, Trust, Resilience, Governance.

**Definitions:** H: Single entity or small group controls the network, resources, and decision-making. Relies on central authority. M2H: Limited number of influential entities; some distribution of control and decision-making. M: Centralized and decentralized control balance; key functions may be controlled centrally. L2M: Control distributed among more participants; few entities may have notable influence. L: Control and resources widely distributed; network resistant to censorship and single points of failure.

#### 6) Scalability Level

Scalability is a critical aspect of DLT systems, as it impacts the capacity, speed, and overall efficiency of the network. Scalability level influences various aspects of a DLT system, such as Transaction throughput, Latency, Network size, and Resource consumption.

**Definitions:** H: Can support a massive number of transactions per second (tps) with minimal latency, suitable for global applications. M2H: Can handle a significant number of tps with reasonable latency, suitable for large-scale applications. M: Satisfactory transaction throughput and latency; may require improvements for larger applications. L2M: Supports moderate tps with increased latency; suitable for small to medium applications. L: Struggles with low transaction throughput and high latency; not ideal for applications requiring high scalability.

#### 7) Latency Level

Latency is an essential factor in evaluating the performance of a DLT system, as it impacts the responsiveness and overall user experience of the network. Latency level influences various aspects of a DLT system, such as Transaction confirmation time, Block propagation time, Network efficiency, and Consensus convergence time.

**Definitions:** H: Significantly delayed transaction confirmation and block validation, not suitable for real-time or time-sensitive applications. M2H: Slower transaction confirmation and block validation, may not be ideal for time-critical applications. M: Reasonable transaction confirmation and block validation times, adequate for most general applications. L2M: Fast transaction confirmation and block validation, suitable for time-sensitive applications. L: Near-instant transaction confirmation and block validation, ideal for real-time applications and high-frequency trading.

## 8) Cost Level

Cost is an essential factor to consider when evaluating a DLT system, as it impacts the affordability, accessibility, and sustainability of the network. Cost level influences various aspects of a DLT system, such as Energy consumption, Hardware requirements, Network maintenance and Transaction fees.

**Definitions:** H: Significantly high transaction fees and resource consumption, likely to deter frequent transactions or low-value transfers, and may be prohibitive for some applications or users. M2H: Above-average transaction fees and resource consumption, potentially limiting the frequency of transactions or discouraging low-value transfers. M: Average transaction fees and resource consumption, acceptable for various use cases but may not be ideal for high-frequency or low-value transactions. L2M: Low transaction fees and resource consumption, suitable for most applications and users. L: Minimal transaction fees and resource consumption, ideal for frequent transactions and low-value transfers.

## 9) Security Level

Security is a crucial factor to consider when evaluating a DLT system, as it impacts the trustworthiness, stability, and overall integrity of the network. Security level influences various aspects of a DLT system, such as Resistance to attacks, Fault tolerance, Data integrity, and Privacy.

**Definitions:** H: Extremely resistant to attacks and faults, providing the highest level of security and trustworthiness for critical applications. M2H: Strong resistance to attacks and faults, suitable for applications that require a high degree of security and trust. M: Provides a reasonable level of security and attack resistance, appropriate for various use cases but may not be ideal for critical applications. L2M: Limited resistance to attacks and faults, may not be suitable for applications requiring high security or trustworthiness. L: Minimal resistance to attacks and faults, not recommended for applications that require a high level of security or trust.

## 10) Interoperability

Blockchain networks and systems that can communicate, interact, and share data are interoperable. This allows blockchains to efficiently share resources, services, and features. Polkadot, Cosmos, and ICON emphasize compatibility. In the context of DLT and BC, it means providing a layer0 that could serve as a common area between multiple DLTs [96].

**Definitions:** H: A consensus algorithm or blockchain platform with high interoperability can easily communicate with many different blockchain networks and platforms. It supports cross-chain communication and collaboration, enabling data and resource sharing between blockchains. M2H: A consensus algorithm or blockchain platform with moderate to high interoperability can interact with many blockchain networks and platforms, but seamless cross-chain communication may require additional development, configuration, or third-party solutions. Its interoperability features may not function with all blockchain networks. M: A consensus algorithm or blockchain platform with moderate interoperability can communicate with a few blockchain networks and platforms. Cross-chain communication may need extensive development, configuration, or third-party solutions and support just certain blockchain networks. L2M: A consensus algorithm or blockchain platform with low to moderate interoperability can interact with a small number of other blockchain networks and platforms, but cross-chain communication may require extensive development, configuration, or third-party solutions. It may support few blockchain networks and have few interoperability features. L: A consensus method or blockchain platform with low interoperability does not permit cross-chain communication and collaboration. It may not function with other blockchain networks without major modification or third-party solutions.

## 11) Complexity

The level of difficulty in comprehending, implementing, and maintaining a system's consensus algorithms, cryptographic methods, protocol layers, and general architecture.

**Definitions:** H: Complex consensus algorithms and blockchain platforms have extensive designs, many components, and sophisticated architecture. Due to its complexity and technological requirements, it may be difficult to understand, implement, and maintain. M2H: A consensus algorithm or blockchain platform with moderate to high complexity has a moderately sophisticated design and architecture. It may require several components or processes for implementation and maintenance and demands a strong understanding of the underlying principles. M: A consensus algorithm or blockchain platform with moderate complexity has a balanced architecture with an average amount of components and operations. It is easy to learn and implement, although some technical knowledge is needed. L2M: A consensus algorithm or blockchain platform with low to moderate complexity has a simpler design and architecture than those

with moderate complexity. Its fewer components and processes make it simpler to understand, implement, and maintain. L: A simple consensus algorithm or blockchain platform has a minimalist design and architecture. Few components and processes make it simple to understand, implement, and maintain. Since the concepts and needs are simple, it's excellent for beginners.

## B. Consensus Based Application (IoT and CPS)

The perfect consensus algorithm would be the best in terms of performance, security, and efficiency across all key measures, and it would be made just for the needs of Cyber-Physical Systems [97] [98]. However, it's important to note that it can be hard to find a consensus method that meets all of these high-level needs. Trade-offs or change on current consensus algorithm to make it work better for your CPS use case [99] [100].

A comparative analysis between all candidates consensus algorithms with the required consensus (ConsensusX) to find the optimal choice for each application among the candidates lists is often required. Figure 16 presents what has been assigned to each attribute to analyze the consensus algorithm candidate lists.

The levels for each attributes for the desirable ConsensusX are based on the reasoning below For a consensus algorithm to be suitable for Cyber Physical Systems (CPS), it should exhibit the discussed desirable levels for attributes in Table XLII

TABLE XLII  
CONSENSUSX ALGORITHM LEVEL ASSIGNING AND WHY?

Attributes	Assigned Level	Reason
Hardware Requirements	Low to Moderate (L2M)	CPS systems should be cost-effective and capable of running on various hardware without being resource-intensive.
Pre-Trust Level	Low to Moderate (L2M)	Trust between network at least initial users should be established to ensure secure and efficient operations, However, some level of trustlessness is still desirable.
Tolerance Level	Moderate to High (M2H)	The consensus algorithm should be able to tolerate a number of faulty or malicious nodes without compromising the network's integrity and operations.
Overhead Computation	Low (L)	Computation overhead should be minimized to ensure efficient transaction processing and scalability a CPS.
Centralization Level	Low to Moderate (L2M)	Decentralization is essential to prevent single points of failure and enhance network resilience.
Scalability Level	High (H)	The consensus algorithm should support high volume of transaction, as CPS systems need to handle a large number of transactions in an efficient way.
Latency Level	Low (L)	Fast transaction confirmations are crucial in CPS systems to enable real-time decision-making.
Cost Level	Low to Moderate (L2M)	The consensus algorithm should be cost-efficient to implement and maintain, without having significant operational expenses.
Security Level	High (H)	A secure consensus algorithm is essential to protect, and due to the secure nature of DLTs, security required at its highest to protect data from unauthorized access or tampering.
Interoperability	High (H)	Integration and data exchange with other DLT systems and CPS infrastructure is crucial. The consensus algorithm should support a high level of interoperability.
Complexity Level	Low to Moderate (L2M)	The consensus algorithm should relatively be simple to implement, understand, and manage for efficient operations in CPS.

## VII. Evaluation of the Algorithms based on the Methodology

In this section, evaluation of all candidate consensus algorithms is made using the methodology in section VI to minimize the list and reach to a list of consensus algorithms that are relatively suitable to CPS. Tables XLIII, XLIV, XLV, and XLVI represent the comparison for candidates' attributes and ConsensusX assigned attributes.

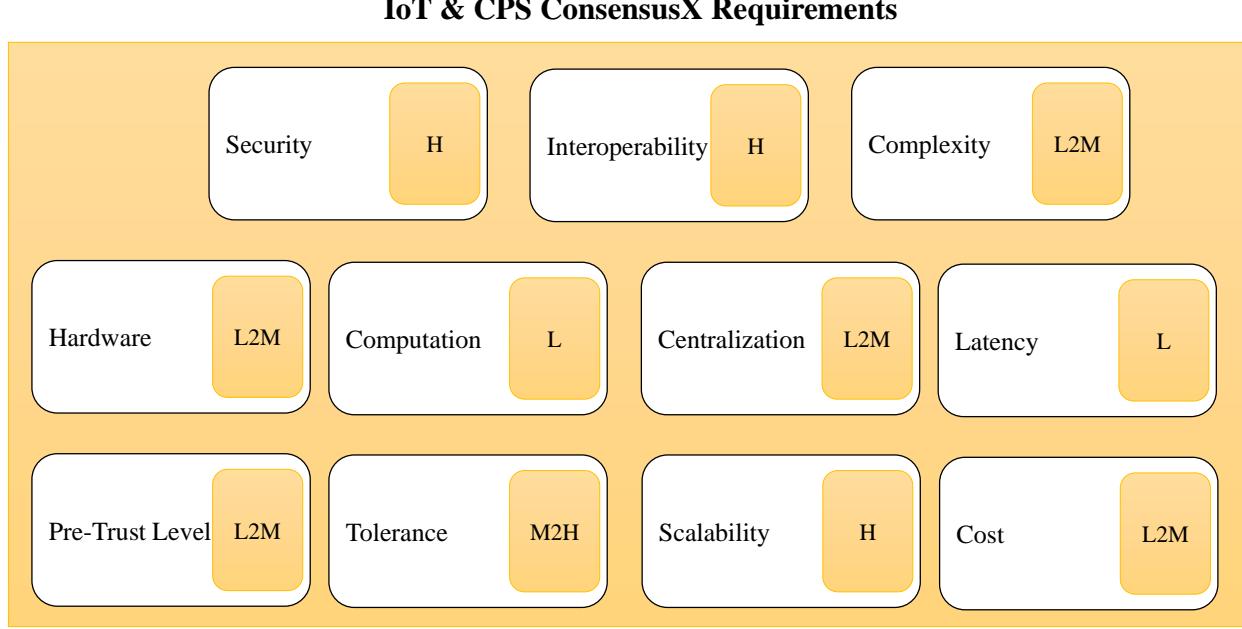


Fig. 16. Internet of Things and Cyber Physical Systems desirable ConsensusX Algorithm Attributes.

#### A. Consensus Algorithms Candidates Lists Versus ConsensusX

TABLE XLIII  
CANDIDATES LIST 1.

Attributes	Hard.	Trust	Toler.	Compu.	Centra.	Scala.	late.	Cost	Secu.	Interoper.	Compl.
PoS	M	M	M	M	M	M	M	M	H	M	M
DPoS	L	M2H	M	L2M	M2H	H	L	L	M2H	M	M
LPoS	M	M	M	M	M	M	M	M	H	M	M
PPoS	M	M	M	M	L	H	L	M	H	M	M
NPoS	M	M	M	M	M	H	L	M	H	M	M
BPoS	M	M	M	M	M	M	M	M	H	M	M
Ouroboros	M	M	M	M	L	H	L	M	H	M	M
ConsensusX	L2M	L2M	M2H	L	L2M	H	L	L	H	H	L2M

Figures 17, 18, 19, and 20 illustrate the match level between the target ConsensusX and the candidates. The results indicate that PPoS (Pure Proof of Stake), NPoS (Nominated Proof of Stake), Ouroboros, DPoS (Delegated Proof of Stake), Tendermint, Hashgraph, Avalanche, Tangle, PoA (Proof of Authority), and PoET (Proof of Elapsed Time) have at least 8 attributes matching with ConsensusX which is based on our evaluation are suitable consensus algorithms but not the exact required ConsensusX.

#### B. Discussion

Cyber-Physical Systems (CPS) necessitate consensus algorithms that offer high security, scalability, and performance while consuming little energy and exhibiting minimal latency. PPoS, NPoS, Ouroboros, DPoS, Tendermint, Hashgraph, Avalanche, Tangle, PoA, and PoET are all acceptable for CPS to varying degrees. DPoS and NPoS, for instance, provide energy efficiency and high throughput, whereas Tendermint and Ouroboros guarantee comprehensive security and decentralization. Hashgraph and Avalanche are promising options due to their scalability and speed of completion, whereas Tangle excels at providing lightweight consensus with minimal resource requirements, making it appropriate for IoT devices. PoA and PoET both take distinctive approaches, with PoA emphasizing trusted authorities for

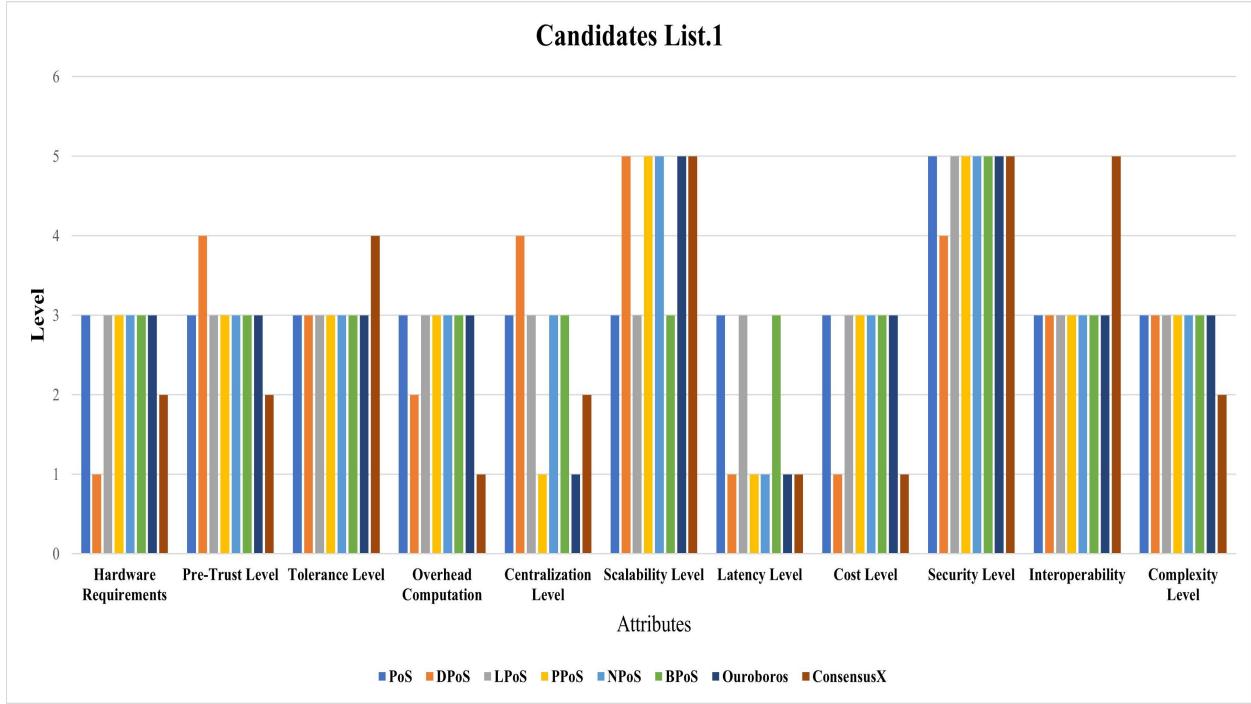


Fig. 17. Candidates List 1 Versus ConsensusX.

TABLE XLIV  
CANDIDATES LIST 2.

Attributes	Hard.	Trust	Toler.	Compu.	Centra.	Scala.	late.	Cost	Secu.	Interoper.	Compl.
<b>BFT</b>	M	M2H	H	M2H	M	M	M	M	H	M	M2H
<b>PBFT</b>	M2H	M2H	H	H	M2H	M	M	M2H	H	M	H
<b>BFA</b>	M	M	H	M2H	M	M	M	M	H	M	M2H
<b>IBFT</b>	M2H	M2H	H	H	M2H	M	M	M2H	H	M	H
<b>HBBFT</b>	M2H	M2H	H	H	M2H	M	M	M2H	H	M	H
<b>RCPA</b>	M	M2H	H	M2H	M	M2H	M2H	M	H	M2H	M2H
<b>Tendermint</b>	M	M	H	M	M	H	M2H	M	H	M2H	M
<b>Hashgraph</b>	M	M	H	M	L	H	H	M	H	M2H	M
<b>Avalanche</b>	M	M	H	M	L	H	H	M	H	M2H	M
<b>ConsensusX</b>	L2M	L2M	M2H	L	L2M	H	L	L	H	H	L2M

TABLE XLV  
CANDIDATES LIST 3.

Attributes	Hard.	Trust	Toler.	Compu.	Centra.	Scala.	late.	Cost	Secu.	Interoper.	Compl.
<b>PoC</b>	M	M	M2H	M	M	M	M	M	M2H	M	M
<b>PoI</b>	M	M2H	M2H	M	M2H	M2H	M2H	M	M2H	M	M
<b>PoCon</b>	M	M	M2H	M	M	M2H	M2H	M	M2H	M	M
<b>PoR</b>	M	M2H	M2H	M	M2H	M2H	M2H	M	M2H	M	M
<b>PoW</b>	H	L	M2H	H	H	L	M	H	H	M	H
<b>Tangle</b>	L	L	H	L	L	H	M	L	H	M2H	M
<b>PoA</b>	M	H	H	L	M2H	H	M2H	L	H	M2H	M
<b>PoAct</b>	M	M	M2H	M	M	M	M	M	M2H	M	M
<b>ConsensusX</b>	L2M	L2M	M2H	L	L2M	H	L	L	H	H	L2M

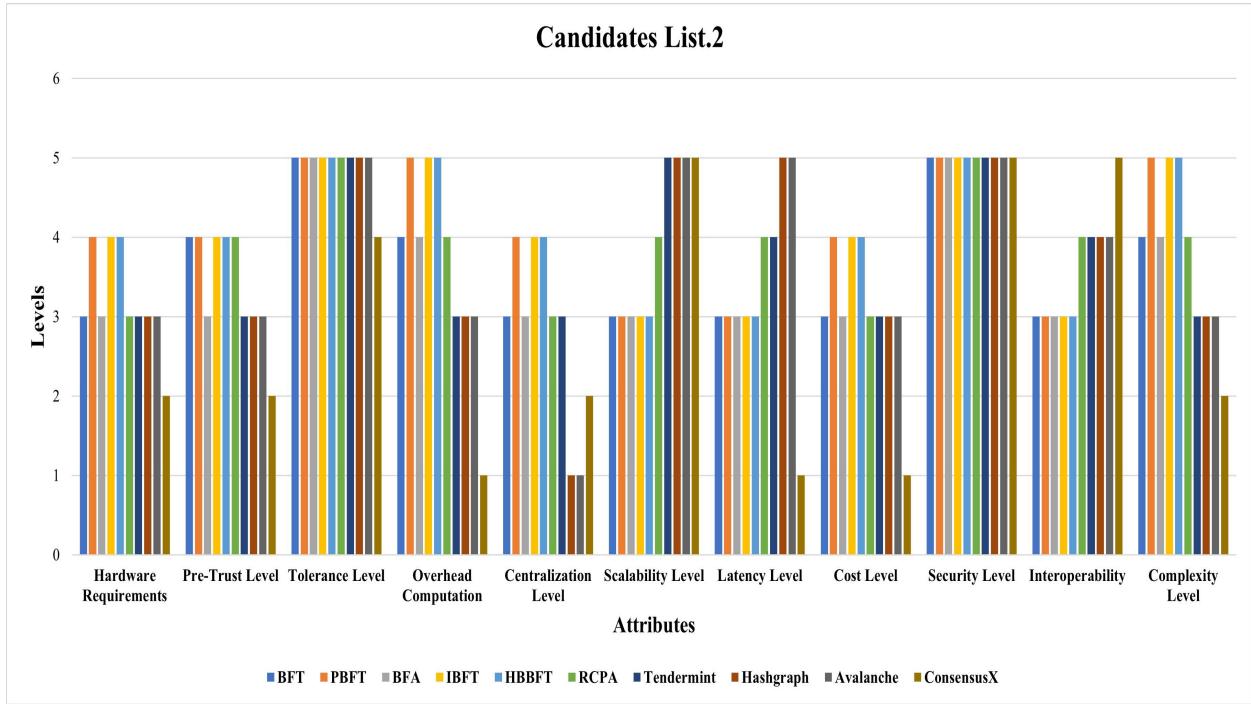


Fig. 18. Candidates List 2 Versus ConsensusX.

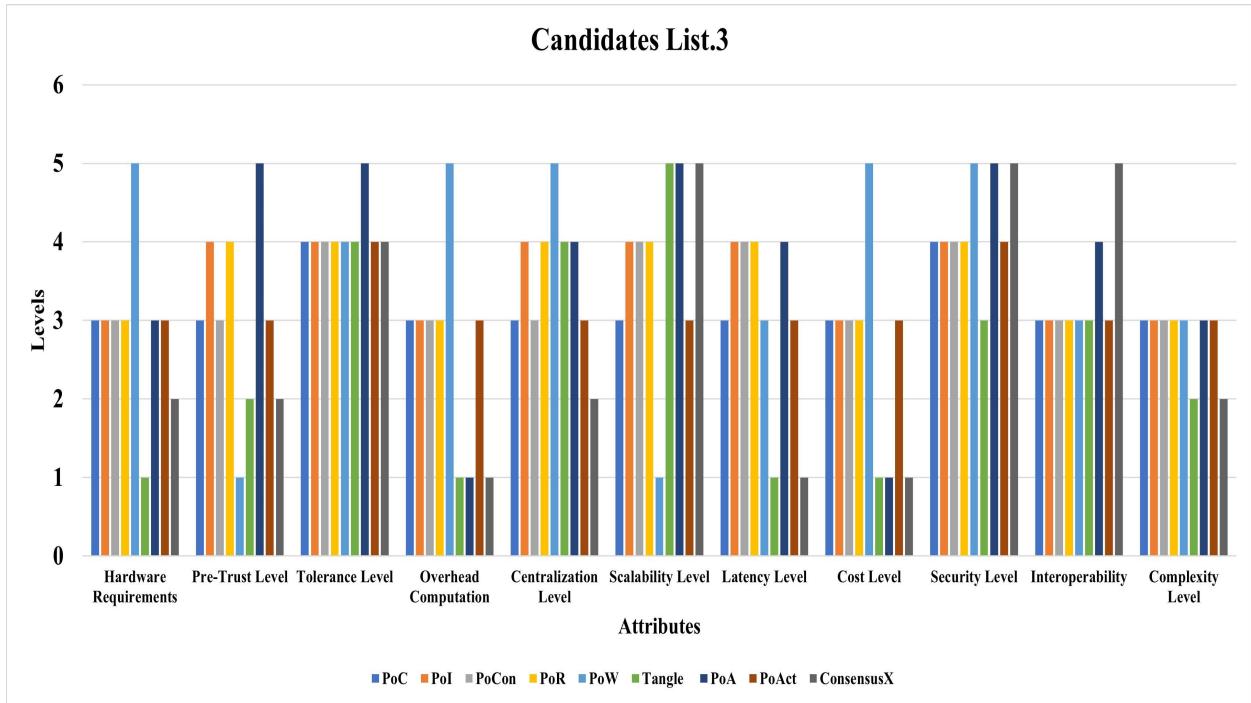


Fig. 19. Candidates List 3 Versus ConsensusX.

TABLE XLVI  
CANDIDATES LIST 4.

Attributes	Hard.	Trust	Toler.	Compu.	Centra.	Scala.	Late.	Cost	Secu.	Interoper.	Compl.
<b>CFT</b>	M	M2H	M2H	M	M2H	M	M	M	M2H	M	M
<b>Paxos</b>	M	M2H	M2H	M	M2H	M	M	M	M2H	M	M
<b>Raft</b>	M	M2H	M2H	M	M2H	M	M	M	M2H	M	M
<b>PoET</b>	L	M	M2H	L	M	M	M	L	M	M	L
<b>PoB</b>	M	M	M2H	M	M2H	M	M	M	M2H	M	M
<b>PoCov</b>	M	M	M2H	M	M	M	M	M	M2H	M	M
<b>ConsensusX</b>	L2M	L2M	M2H	L	L2M	H	L	L	H	H	L2M

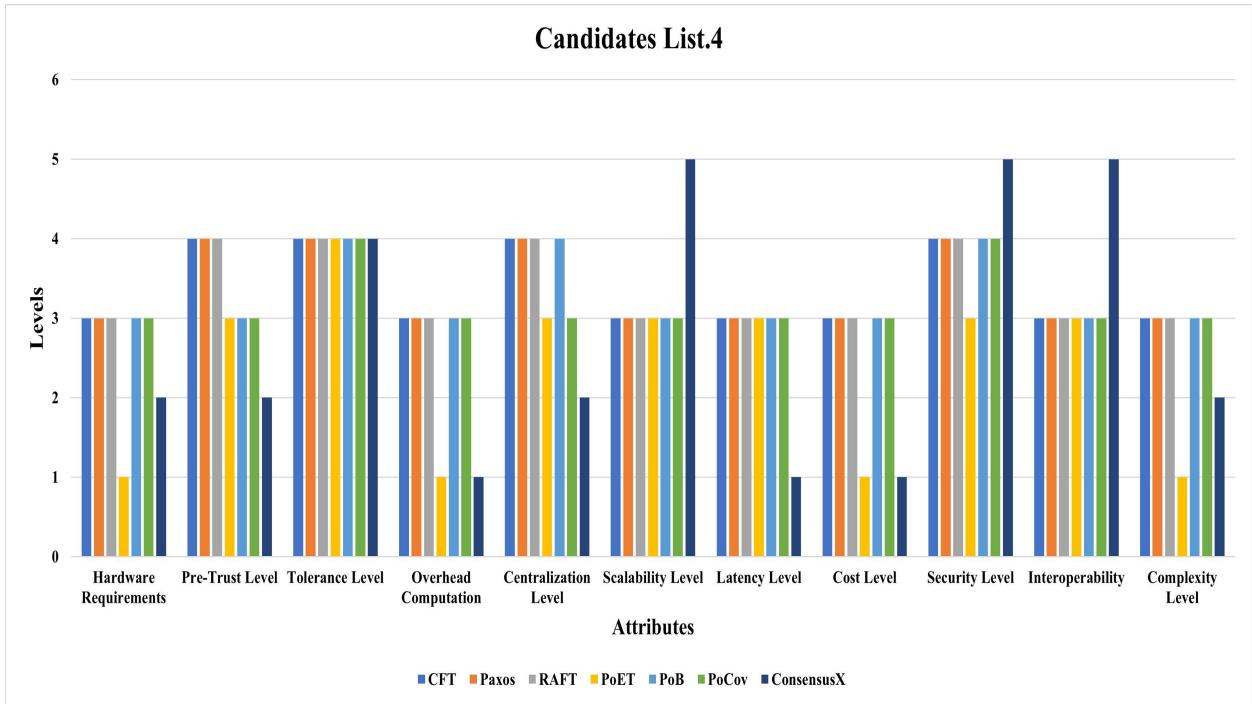


Fig. 20. Candidates List 4 Versus ConsensusX.

decision-making and PoET utilizing hardware-based random leader selection for impartiality and energy efficiency. All the listed CAs have similar or close levels for more than 7 attributes out of 11. It is obviously very hard to find the perfect match to certain requirement for a specific field from a pre-established CAs. From the analysis above, PPoS, NPoS, Ouroboros, DPoS, Tendermint, Hashgraph, Avalanche, Tangle, PoA, and PoET are the closest to be used in a CPS environment. Each CA has been used in a certain ledger structure that could impact the overall performance for each one.

- Algorithm for PPoS (Pure Proof of Stake): To select validators, this consensus algorithm employs a blockchain-based ledger structure and a novel cryptographic sorting procedure.
- NPoS (Nominated Proof of Stake) - Polkadot: NPoS employs a blockchain-based ledger structure featuring a Relay Chain, Parachains, and Bridges to connect multiple blockchains.
- Ouroboros - Cardano: Ouroboros also utilizes a blockchain-based ledger structure, with epochs divided into slots and slot leaders selected via a secure multiparty computation protocol.
- DPoS (Delegated Proof of Stake) - EOS, Bitshares: DPoS operates on a blockchain-based ledger structure, with a limited number of elected delegates validating transactions and producing blocks.
- Tendermint - Cosmos: Tendermint is a BFT-based consensus algorithm employing a blockchain-based ledger structure, with a modular architecture that supports multiple application-specific blockchains connected via the Cosmos Hub.

- Hedera is the hashgraph for the Hedera tree Hashgraph uses a directed acyclic graph (DAG) structure, known as the hashgraph, to record transactions and reach consensus via a virtual voting mechanism.
- Avalanche - Avalanche Protocol: Avalanche consensus utilizes a directed acyclic graph (DAG) structure with multiple subnets and chains, as well as a novel probabilistic sampling technique for validator selection.
- Tangle - IOTA: Tangle also employs a directed acyclic graph (DAG) structure, in which transactions are connected via a web-like structure as opposed to a linear blockchain.
- PoA (Proof of Authority) - VeChain, POA Network: PoA typically uses a blockchain-based ledger structure with a limited number of trusted validators who have the authority to create and validate blocks based on their reputation.
- Hyperledger Sawtooth's PoET (Proof of Elapsed Time) implementation: PoET utilizes a blockchain-based ledger structure and a novel method that utilizes secure enclaves in hardware to randomly select block producers.

TABLE XLVII  
EXISTED IMPLEMENTATIONS FOR THE FINAL CANDIDATES LIST WITH HIGHER SUITABILITY TO CPS.

Implementations	Pros	Cons
PPoS (Algorand)	High security, decentralization, energy efficiency	Complex cryptographic processes, scalability challenges
NPoS (Polkadot)	Scalability, interoperability, efficient consensus	Complex architecture, centralization risk
Ouroboros (Cardano)	Energy efficiency, strong security, adaptability	Complexity, potential latency, evolving protocol
DPoS (EOS, Bit-shares)	High performance, scalability, reduced energy consumption	Centralization risk, voter apathy, reliance on delegates
Tendermint (Cosmos)	High throughput, modularity, interoperability	Centralization risk, complex architecture, reliance on validators
Hashgraph (Hedera Hashgraph)	Fast transaction processing, high throughput, asynchronous BFT	Limited decentralization, patented technology, complex data structure
Avalanche	High scalability, robust security, adaptable architecture	New protocol, complex validator selection, unproven long-term stability
Tangle (IOTA)	Scalability, zero transaction fees, IoT suitability	Lack of robust security, reliance on coordinator nodes, complex data structure
PoA (VeChain, POA Network)	High performance, reduced energy consumption, predictable validator selection	Centralization risk, reliance on trusted authorities, limited decentralization
PoET (Hyperledger Sawtooth)	Fairness, energy efficiency, hardware-based security	Dependence on specific hardware, scalability challenges, limited applicability outside enterprise use cases

### VIII. Conclusion

In this study, 30 consensus algorithms, including PPoS, NPoS, Ouroboros, DPoS, Tendermint, Hashgraph, Avalanche, Tangle, PoA, and PoET, were investigated. Eleven attributes, including security, scalability, decentralization, energy efficiency, and performance, were used to evaluate each algorithm based on the requirements and trade-offs of cyber-physical systems. In addition, a detailed comparison of PoS family algorithms, BFT family algorithms, weight family, and others, as well as the benefits and drawbacks of each consensus algorithm based on their ledger structures, were provided. While each consensus algorithm has its own strengths and limitations, it is crucial to consider the use case's specific requirements and constraints when selecting an appropriate consensus mechanism. In addition, it is essential to investigate potential modifications and adaptations of these algorithms to address their limitations and difficulties. In the end, the selection of a consensus algorithm will hinge on the optimal balance between the desired level of security, performance, decentralization, and other application-specific factors. The work resulted in a list of candidates with greater compatibility with CPS and their existing implementations, as listed in Table XLVII.

### References

- [1] F. B. Insights, "Blockchain in retail market size, share & industry analysis, by component (platform, services), by provider (application and solution provider, middleware provider, infrastructure and protocol provider), by organization size (large enterprises, small & medium enterprises) others and regional forecast, 2019-2026," <https://www.fortunebusinessinsights.com/block-chain-in-retail-market-102092>, 2020.

- [2] R. Kamath, "Food traceability on blockchain: Walmart's pork and mango pilots with ibm," *The Journal of the British Blockchain Association*, vol. 1, pp. 1–12, 07 2018.
- [3] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari, and Y. Cao, "A survey on blockchain technology: Evolution, architecture and security," *IEEE Access*, vol. 9, pp. 61 048–61 073, 2021.
- [4] Y. Xiong and C. Hu, "Comparative research on blockchain consensus algorithms," in *International Conference on Blockchain Technology and Information Security (ICBCTIS)*, 2022, pp. 160–164.
- [5] X. Li, Q. Zhu, N. Qi, J. Huang, Y. Yuan, and F.-Y. Wang, "Blockchain consensus algorithms: A survey," in *China Automation Congress (CAC)*, 2021, pp. 4053–4058.
- [6] A. Panwar and V. Bhatnagar, "Distributed ledger technology (DLT): The beginning of a technological revolution for blockchain," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, 2020, pp. 1–5.
- [7] K. Kant, S. Pandey, and U. Shanker, "A journey from commit processing in distributed databases to consensus in blockchain," in *IEEE 38th International Conference on Data Engineering (ICDE)*, 2022, pp. 3236–3240.
- [8] 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 Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, Jul 2018.
- [9] L. Rachakonda, A. K. Bapatla, S. P. Mohanty, and E. Kougianos, "SaYoPillow: Blockchain-Integrated Privacy-Assured IoMT Framework for Stress Management Considering Sleeping Habits," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 1, pp. 20–29, Feb 2021.
- [10] S. L. T. Vangipuram, S. P. Mohanty, E. Kougianos, and C. Ray, "agroString: Visibility and Provenance through a Private Blockchain Platform for Agricultural Dispense towards Consumers," *Sensors*, vol. 22, no. 21, p. 8227, Oct 2022. [Online]. Available: <https://doi.org/10.3390/s22218227>
- [11] A. Mitra, S. L. T. Vangipuram, A. K. Bapatla, V. K. V. V. Bathalapalli, S. P. Mohanty, E. Kougianos, and C. Ray, "Everything You wanted to Know about Smart Agriculture," *CoRR*, vol. abs/2201.04754, 2022. [Online]. Available: <https://arxiv.org/abs/2201.04754>
- [12] A. K. Bapatla, S. P. Mohanty, E. Kougianos, D. Puthal, and A. Bapatla, "PharmaChain: A blockchain to ensure counterfeit-free pharmaceutical supply chain," *IET Networks*, vol. 12, no. 2, pp. 53–76, Mar 2023.
- [13] M. Kaur and S. Gupta, "Blockchain consensus protocols: State-of-the-art and future directions," in *International Conference on Technological Advancements and Innovations (ICTAI)*, 2021, pp. 446–453.
- [14] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Consensus in the age of blockchains," *arXiv preprint arXiv:1711.03936*, 2017.
- [15] R. Zhang and B. Preneel, "On the necessity of a prescribed block validity consensus: Analyzing bitcoin unlimited mining protocol," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1661–1666.
- [16] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International Workshop on Open Problems in Network Security*. Springer, Cham, 2015, pp. 112–125.
- [17] L. Xin, L. Jing, and W. Jian, "Research on consensus algorithms of blockchain," in *Proceedings of the 3rd International Conference on Big Data Research*, pp. 37–41.
- [18] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," National Institute of Standards and Technology, Tech. Rep. NISTIR 8202, 2018.
- [19] Y. Wang and J. Wang, "A survey of consensus algorithms for blockchain," in *International Conference on Big Data Engineering*. Springer, Singapore, 2019, pp. 299–310.
- [20] G. T. Nguyen, K. Kim, and S. Lee, "A survey about consensus algorithms used in blockchain," *Journal of Information Processing Systems*, vol. 15, no. 1, pp. 1–20, 2019.
- [21] J. Merkle and F. Niebling, "A comparative analysis of blockchain consensus algorithms," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.
- [22] R. Abbasi, N. Naseer, and M. Gidlund, "A survey on consensus mechanisms and their applications in blockchain for iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 6929–6954, 2021.
- [23] L. Garg and A. Choudhary, "Blockchain consensus algorithms: A comprehensive review," *Journal of Network and Computer Applications*, vol. 180, p. 103019, 2021.
- [24] I. R. Pustokhina, D. A. Pustokhin, and V. V. Strijov, "Consensus algorithms in blockchain: Comparative analysis and classification," *Expert Systems with Applications*, vol. 153, p. 113409, 2020.
- [25] S. Soleymani and F. Anjomshoa, "A comprehensive study of blockchain consensus algorithms," in *6th International Conference on Web Research (ICWR)*. IEEE, 2020, pp. 83–88.
- [26] X. Deng, K. Li, Z. Wang, J. Li, and Z. Luo, "A survey of blockchain consensus algorithms," in *International Conference on Blockchain Technology and Information Security (ICBCTIS)*, 2022, pp. 188–192.
- [27] J. M. Jose and P. V, "A survey on consensus algorithms in blockchain based on post quantum cryptosystems," in *5th International Conference on Computational Intelligence and Networks (CINE)*, 2022, pp. 1–6.
- [28] H. Ju, X. Zhang, H. Jia, X. Zhang, E. Zhu, K. Yan, and J. Guo, "A survey on efficient consensus mechanism for electricity information acquisition system," in *9th International Conference on Smart Grid (icSmartGrid)*, 2021, pp. 124–127.

- [29] W. Gu, J. Li, and Z. Tang, "A survey on consensus mechanisms for blockchain technology," in *International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*, 2021, pp. 46–49.
- [30] S. Islam, M. J. Islam, M. Hossain, S. Noor, K.-S. Kwak, and S. M. Riazul Islam, "A survey on consensus algorithms in blockchain-based applications: Architecture, taxonomy, and operational issues," *IEEE Access*, pp. 1–1, 2023.
- [31] Y. Guichun, F. Liuping, N. Jing, and Y. Xiaoyu, "Survey on the application of blockchain in digital rights protection," in *International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, 2020, pp. 183–187.
- [32] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P. K. Singh, and W.-C. Hong, "A survey on decentralized consensus mechanisms for cyber physical systems," *IEEE Access*, vol. 8, pp. 54 371–54 401, 2020.
- [33] L. Li, P. Shi, X. Fu, P. Chen, T. Zhong, and J. Kong, "Three-dimensional tradeoffs for consensus algorithms: A review," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1216–1228, 2022.
- [34] M. Rwibasira and R. Suchithra, "A survey paper on consensus algorithm of mobile-healthcare in blockchain network," in *International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1–5.
- [35] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. Rehmani, "Applications of blockchains in the internet of things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2019, 2nd Quart.
- [36] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53 019–53 033, 2018.
- [37] D. Puthal and S. P. Mohanty, "Proof of authentication: IoT-friendly blockchains," *IEEE Potentials Magazine*, vol. 38, no. 1, pp. 26–29, 2019.
- [38] D. Puthal, S. P. Mohanty, V. P. Yanambaka, and E. Koulianios, "PoAh: A novel consensus algorithm for fast scalable private blockchain for large-scale IoT frameworks," 2020, arXiv preprint arXiv:2001.07297. [Online]. Available: <https://arxiv.org/pdf/2001.07297.pdf>
- [39] A. K. Bapatla, S. P. Mohanty, E. Koulianios, D. Puthal, and A. Bapatla, "Pharmachain: A blockchain to ensure counterfeit-free pharmaceutical supply chain," *IET Networks*, vol. 12, no. 2, pp. 53–76, 2023. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ntw2.12041>
- [40] B. S. Egala, A. K. Pradhan, P. Dey, V. Badarla, and S. P. Mohanty, "Fortified-chain 2.0: Intelligent blockchain for decentralized smart healthcare system," *IEEE Internet of Things Journal*, pp. 1–1, 2023.
- [41] P. R. Nair and D. R. Dorai, "Evaluation of performance and security of proof of work and proof of stake using blockchain," in *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 279–283.
- [42] S. Talukder and R. Vaughn, "A template for alternative proof of work for cryptocurrencies," in *International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 2021, pp. 1–6.
- [43] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.
- [44] R. Beer and T. Sharma, "A quick look at cryptocurrency mining: Proof of work," in *2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, vol. 2, 2022, pp. 651–656.
- [45] A. Back, "Hashcash- a denial of service counter-measure," *cypherspace*, Aug. 2002. [Online]. Available: <chrome-extension://efaidnbmnnibpcajpcgclefindmkaj/http://www.hashcash.org/papers/hashcash.pdf>
- [46] S. Yan, "Analysis on blockchain consensus mechanism based on proof of work and proof of stake," in *International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACA)*, 2022, pp. 464–467.
- [47] S. Popov, "The tangle," 2016, white paper. [Online]. Available: [https://iota.org/IOTA\\_Whitepaper.pdf](https://iota.org/IOTA_Whitepaper.pdf)
- [48] P. R. Nair and D. R. Dorai, "Evaluation of performance and security of proof of work and proof of stake using blockchain," in *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 279–283.
- [49] S. M. S. Saad, R. Z. R. M. Radzi, and S. H. Othman, "Comparative analysis of the blockchain consensus algorithm between proof of stake and delegated proof of stake," in *International Conference on Data Science and Its Applications (ICoDSA)*, 2021, pp. 175–180.
- [50] E. Androulaki *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," 2018, available: <https://arxiv.org/pdf/1801.10228.pdf>.
- [51] Y. Du, Z. Wang, J. Li, L. Shi, D. N. K. Jayakody, Q. Chen, W. Chen, and Z. Han, "Blockchain-aided edge computing market: Smart contract and consensus mechanisms," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.
- [52] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," 2017, white paper. [Online]. Available: <https://arxiv.org/pdf/1607.01341.pdf>
- [53] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," 2017, white paper. [Online]. Available: <https://eprint.iacr.org/2016/889.pdf>
- [54] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," 2016, white paper. [Online]. Available: <https://polkadot.network/PolkaDotPaper.pdf>
- [55] J. Kwon and E. Buchman, "Cosmos: A network of distributed ledgers," 2016, white paper. [Online]. Available: <https://cosmos.network/resources/whitepaper>

- [56] J. Pan, Z. Song, and W. Hao, “Development in consensus protocols: From pow to pos to dpos,” in *2nd International Conference on Computer Communication and Network Security (CCNS)*, 2021, pp. 59–64.
- [57] L. M. Bach, B. Mihaljevic, and M. Zagar, “Comparative analysis of blockchain consensus algorithms,” in *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018, pp. 1545–1550.
- [58] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in *IEEE International Congress on Big Data*. IEEE, 2017, pp. 557–564.
- [59] M. Castro and B. Liskov, “Practical byzantine fault tolerance,” in *OSDI*, 1999, pp. 173–186.
- [60] M. Vukolić, “The quest for scalable blockchain fabric: Proof-of-work vs. bft replication,” in *International Workshop on Open Problems in Network Security*. Springer, 2015, pp. 112–125.
- [61] Y. Mao and F. P. Junqueira, “The state of the art in the design and implementation of bft consensus algorithms,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 72–109, 2018.
- [62] K. Hu, K. Guo, Q. Tang, Z. Zhang, H. Cheng, and Z. Zhao, “Leopard: Towards high throughput-preserving bft for large-scale systems,” in *IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, 2022, pp. 157–167.
- [63] M. Castro and B. Liskov, “Practical byzantine fault tolerance,” in *Proc. 3rd Symp. Operating Syst. Design Implement.*, New Orleans, LA, USA, 1999.
- [64] D. Mazieres, “The stellar consensus protocol: A federated model for internet-level consensus,” 2015, white paper. [Online]. Available: <https://www.stellar.org/papers/stellar-consensus-protocol.pdf>
- [65] R. Saltini, “IBFT liveness analysis,” in *IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 245–252.
- [66] J. Morgan, “Quorum: A permissioned implementation of ethereum supporting data privacy,” 2016, white paper. [Online]. Available: <https://github.com/ethereum/wiki/wiki/Quorum-Whitepaper>
- [67] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, “The honey badger of bft protocols,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 31–42.
- [68] D. Schwartz, N. Youngs, and A. Britto, “The ripple protocol consensus algorithm,” 2014, white paper. [Online]. Available: [https://ripple.com/files/ripple\\_consensus\\_whitepaper.pdf](https://ripple.com/files/ripple_consensus_whitepaper.pdf)
- [69] E. Buchman, “Tendermint: Byzantine fault tolerance in the age of blockchains,” 2016, white paper. [Online]. Available: <https://arxiv.org/pdf/1607.06450.pdf>
- [70] L. Baird, “The swirls hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance,” 2016, white paper. [Online]. Available: <https://www.swirls.com/downloads/SWIRLDS-TR-2016-01.pdf>
- [71] T. Rocket, “Avalanche: A novel metastable consensus protocol family for cryptocurrencies,” 2018, white paper. [Online]. Available: <https://ipfs.io/ipfs/QmUy4jh5mGNZvLkjies1RWM4YuvJh5o2FYopNPVYwrRVGV>
- [72] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, 5th ed. Addison-Wesley, 2011.
- [73] A. Goel and K. A. Sakallah, “Towards an automatic Proof of Lamport’s Paxos,” in *Formal Methods in Computer Aided Design (FMCAD)*, 2021, pp. 112–122.
- [74] B. Li and J. Jiang, “Security analysis of paxos mechanism design based on game theory,” in *IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, vol. 1, 2020, pp. 59–66.
- [75] Y. Wu, Y. Wu, Y. Liu, and T. Shi, “The research of the optimized solutions to raft consensus algorithm based on a weighted pagerank algorithm,” in *Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, 2022, pp. 784–789.
- [76] D. Huang, X. Ma, and S. Zhang, “Performance analysis of the raft consensus algorithm for private blockchains,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 172–181, 2020.
- [77] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, “On security analysis of proof-of-elapsed-time (poet),” in *Proc. Int. Symp. Stabilization, Saf., Secur. Distrib. Syst.* Cham, Switzerland: Springer, 2017.
- [78] 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,” 2018, available: <http://arxiv.org/abs/1805.02707>.
- [79] S. Developers, “Slimcoin: The first cryptocurrency with proof of burn,” 2014. [Online]. Available: <https://github.com/slimcoin-project/Slimcoin/blob/master/README.md>
- [80] M. Bartoletti and L. Pompianu, “An empirical analysis of the distribution of wealth in the bitcoin network,” *arXiv preprint arXiv:1702.02867*, 2017.
- [81] C. Developers, “Counterparty,” 2014. [Online]. Available: <https://counterparty.io/>
- [82] R. Smith, “Proof of burn | consensus through coin destruction,” Article Published by Coin Central, 2019, available: <https://coincentral.com/proof-of-burn/>.
- [83] H. S. Inc, “The helium whitepaper: A peer-to-peer wireless network, built on the blockchain,” 2019, white paper. [Online]. Available: <https://helium.com/pdf/helium-whitepaper.pdf>
- [84] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, “Proofs of space,” *Advances in Cryptology*, pp. 585–605, 2015.
- [85] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, “Proofs of space,” in *Proc. 35th Annu. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA, Aug 2015, pp. 585–605.

- [86] S. Park, S. Kim, Y. Kim, S. Kim, J. Kim, and S. Kim, “Chained proof-of-stake: A new secure, scalable blockchain consensus protocol,” *arXiv preprint arXiv:1804.07391*, 2018.
- [87] A. Shoker, “Sustainable blockchains through proof of exercise,” in *International Conference on Cloud Computing and Services Science*. Springer, 2017, pp. 361–372.
- [88] NEM, “Investor harvesting: Proof-of-importance,” Accessed: Apr. 22, 2019, NEM (XEM), 2018, available: <https://nem.io/xem/harvesting-and-poi/>.
- [89] Q. Zhuang, Y. Liu, L. Chen, and Z. Ai, “Proof of reputation: A reputation-based consensus protocol for blockchain based systems,” 07 2019, pp. 131–138.
- [90] H. Song, N. Zhu, R. Xue, J. He, K. Zhang, and J. Wang, “Proof-of-contribution consensus mechanism for blockchain and its application in intellectual property protection,” *Information Processing and Management*, vol. 58, no. 3, p. 102507, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306457321000170>
- [91] V. Foundation, “Vechain whitepaper,” 2018. [Online]. Available: [https://cdn.vechain.com/vechainthor\\_development\\_plan\\_and\\_whitepaper\\_en\\_v1.0.pdf](https://cdn.vechain.com/vechainthor_development_plan_and_whitepaper_en_v1.0.pdf)
- [92] D. Project, “Decred: A self-funding cryptocurrency,” 2018, white paper. [Online]. Available: <https://decred.org/dtb001.pdf>
- [93] S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, “PUFchain: Hardware-assisted blockchain for sustainable simultaneous device and data security in internet of everything (IoE),” *IEEE Consumer Electronics Magazine (MCE)*, vol. 9, no. 2, pp. 8–16, 2020. [Online]. Available: <https://doi.org/10.1109/MCE.2020.2965611>
- [94] V. K. V. Bathalapalli, S. P. Mohanty, E. Kougianos *et al.*, “PUFchain 2.0: Hardware-assisted robust blockchain for sustainable simultaneous device and data security in smart healthcare,” *SN Computer Science*, vol. 3, no. 2, p. 344, 2022. [Online]. Available: <https://doi.org/10.1007/s42979-022-01238-2>
- [95] 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 of Things Journal (JIoT)*, vol. 7, no. 3, pp. 2343–2355, 2020. [Online]. Available: <https://doi.org/10.1109/JIoT.2019.2953714>
- [96] A. Alkhodair, S. P. Mohanty, and E. Kougianos, “Flexichain 2.0: Nodechain assisting integrated decentralized vault for effective data authentication and device integrity in complex cyber-physical systems,” *arXiv Computer Science arXiv:2304.08713*, p. 25, Apr 2023.
- [97] A. Alkhodair, S. Mohanty, E. Kougianos, and D. Puthal, “McPoRA: A multi-chain proof of rapid authentication for post-blockchain based security in large scale complex cyber-physical systems,” in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2020, pp. 446–451.
- [98] A. J. Alkhodair, S. P. Mohanty, and E. Kougianos, “ASID: Accessible secure unique identification file based device security in next generation blockchains,” in *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1–2.
- [99] A. J. Alkhodair, S. P. Mohanty, and E. Kougianos, “Flexichain: A minerless scalable next generation blockchain for rapid data and device security in large scale complex cyber-physical systems,” *SN Computer Science*, vol. 3, no. 3, p. 235, 2022. [Online]. Available: <https://doi.org/10.1007/s42979-022-01139-4>
- [100] A. Alkhodair, S. P. Mohanty, and E. Kougianos, “Flexichain 3.0: Distributed ledger technology-based intelligent transportation for vehicular digital asset exchange in smart cities,” *Sensors*, vol. 23, no. 8, 2023. [Online]. Available: <https://www.mdpi.com/1424-8220/23/8/4114>



**Ahmad J. Alkhodair** received a bachelor’s degree (Honors) in computer engineering from Fahad bin Sultan University (FBSU), Saudi Arabia-Tabuk, in 2012 and master’s in computer engineering in 2017 from Denver University (DU), Colorado-Denver, USA. A Faculty Member in University of Tabuk, Saudi Arabia-Tabuk at the department of Computer Engineering. Currently a Ph.D. candidate in the research group at Smart Electronics Systems Laboratory (SESL) at Computer Science and Engineering at the University of North Texas, Denton, TX. Research interests include Distributed Ledger Technology (DLT), Cyber Physical Systems (CPS), Artificial Intelligence (AI) in Smart Cities.



**Saraju P. Mohanty** received the bachelor's degree (Honors) in electrical engineering from the Orissa University of Agriculture and Technology, Bhubaneswar, in 1995, the master's degree in Systems Science and Automation from the Indian Institute of Science, Bengaluru, in 1999, and the Ph.D. degree in Computer Science and Engineering from the University of South Florida, Tampa, in 2003. He is a Professor with the University of North Texas. His research is in "Smart Electronic Systems" which has been funded by National Science Foundations (NSF), Semiconductor Research Corporation (SRC), U.S. Air Force, IUSSTF, and Mission Innovation. He has authored 450 research articles, 5 books, and 9 granted and pending patents. His Google Scholar h-index is 54 and i10-index is 227 with 12,000 citations. He is regarded as a visionary researcher on Smart Cities technology in which his research deals with security and energy aware, and AI/ML-integrated smart components. He introduced the Secure Digital Camera (SDC) in 2004 with built-in security features designed using Hardware Assisted Security (HAS) or Security by Design (SbD) principle. He is widely credited as the designer for the first digital watermarking chip in 2004 and first the low-power digital watermarking chip in 2006. He is a recipient of 16 best paper awards, Fulbright Specialist Award in 2020, IEEE Consumer Electronics Society Outstanding Service Award in 2020, the IEEE-CS-TCVLSI Distinguished Leadership Award in 2018, and the PROSE Award for Best Textbook in Physical Sciences and Mathematics category in 2016. He has delivered 22 keynotes and served on 14 panels at various International Conferences. He has been serving on the editorial board of several peer-reviewed international transactions/journals, including IEEE Transactions on Big Data (TBD), IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), IEEE Transactions on Consumer Electronics (TCE), and ACM Journal on Emerging Technologies in Computing Systems (JETC). He has been the Editor-in-Chief (EiC) of the IEEE Consumer Electronics Magazine (MCE) during 2016-2021. He served as the Chair of Technical Committee on Very Large Scale Integration (TCVLSI), IEEE Computer Society (IEEE-CS) during 2014-2018 and on the Board of Governors of the IEEE Consumer Electronics Society during 2019-2021. He serves on the steering, organizing, and program committees of several international conferences. He is the steering committee chair/vice-chair for the IEEE International Symposium on Smart Electronic Systems (IEEE-iSES), the IEEE-CS Symposium on VLSI (ISVLSI), and the OITS International Conference on Information Technology (OCIT). He has mentored 2 post-doctoral researchers, and supervised 15 Ph.D. dissertations, 26 M.S. theses, and 20 undergraduate projects.



**Elias Kougianos** received a BSEE from the University of Patras, Greece in 1985 and an MSEE in 1987, an MS in Physics in 1988 and a Ph.D. in EE in 1997, all from Louisiana State University. From 1988 through 1998 he was with Texas Instruments, Inc., in Houston and Dallas, TX. In 1998 he joined Avant! Corp. (now Synopsys) in Phoenix, AZ as a Senior Applications engineer and in 2000 he joined Cadence Design Systems, Inc., in Dallas, TX as a Senior Architect in Analog/Mixed-Signal Custom IC design. He has been at UNT since 2004. He is a Professor in the Department of Electrical Engineering, at the University of North Texas (UNT), Denton, TX. His research interests are in the area of Analog/Mixed-Signal/RF IC design and simulation and in the development of VLSI architectures for multimedia applications. He is an author of over 200 peer-reviewed journal and conference publications.