A Feature Model of Consensus Algorithms in Distributed Ledger Technology

Elena Baninemeh[1](✉), Slinger Jansen[1,2], and Bas Pronk[1]

[1] Information and Computer Science, Utrecht University, Utrecht, The Netherlands
{e.baninemeh,slinger.jansen,b.pronk}@uu.nl
[2] Lappeenranta University of Technology, Lappeenranta, Finland

Abstract. A distributed ledger is a database distributed across multiple systems, with each system holding a synchronized copy of the data. Distributed ledger technology has applications in various healthcare, finance, and cybersecurity domains. However, the intricacies of the features of consensus algorithms, which ensure consistency across different ledgers, remain challenging, as the relevant knowledge is scattered across a wide range of literature or in the form of tacit knowledge of software practitioners. This study presents a systematic data collection comprising an extensive literature review and a set of expert interviews to provide insights into designing and evaluating of consensus algorithms for web3 applications. The usability and usefulness of the extracted knowledge were evaluated by seven experienced practitioners in web3 development companies, resulting in an overview of 13 consensus algorithms, their features, and their impacts on quality models. With this comprehensive knowledge, web3 developers can expedite evaluating, selecting, and implementing consensus algorithms for distributed ledgers.

Keywords: consensus algorithm · algorithm selection · distributed ledger

1 Introduction

Distributed ledger technology (DLT) has emerged as a potential alternative to traditional centralized data management systems. Unlike centralized systems, DLT allows data to be stored and maintained among multiple peers in a network, without relying on a central authority [15]. DLT achieves this through the use of a consensus algorithm that establishes a shared state of the ledger among all network participants. Consensus algorithms are designed to address the challenges of maintaining a distributed ledger, such as ensuring data integrity and preventing malicious attacks [8].

DLT has been applied to various domains, including supply chain management, healthcare, finance, and more [17]. However, DLT is still an emerging technology that faces several significant challenges, including concerns about its

J. M. Machado et al. (Eds.): BLOCKCHAIN 2023, LNNS 778, pp. 264–275, 2023.
https://doi.org/10.1007/978-3-031-45155-3_27

security and scalability [16]. Designing the ledger is a crucial challenge many new DLT projects face, involving making numerous decisions during the design process. The designer must make choices regarding the consensus algorithm, transaction validation mechanism, data storage structure, and access control policies, among other things. These decisions significantly impact the system's security, scalability, and efficiency, which can affect the project's success. Therefore, careful consideration and extensive research must be carried out during the design process to ensure that the DLT project can meet its objectives and deliver optimal performance [20].

Consensus algorithms are a critical component of DLT, as they ensure the consistency of distributed ledgers among network nodes [13]. Due to the wide range of threats that can affect the system, consensus algorithms come in different forms and designs. For instance, large public and cryptocurrency ledgers typically use a proof-of-work algorithm, which requires nodes to solve a complex mathematical puzzle before adding new data. On the other hand, smaller private blockchains often use distributed system consensus algorithms, such as PAXOS and RAFT, which rely on agreement protocols rather than computational puzzles to ensure data consistency [8,13].

In this study, we proposed a systematic approach for collecting data on consensus algorithms to support web3 developers in selecting, creating, and employing consensus algorithms. Our study involved conducting a literature review and interviewing experts to evaluate the usefulness and usability of the extracted knowledge. The study identified 13 consensus algorithms and their features, providing valuable insights into designing and evaluating consensus algorithms.

In Sect. 2 we present the research challenge of capturing knowledge around features of consensus algorithms and propose to do so through literature study and expert interviews. Subsequently, we report on the creation of a feature model in Sect. 3 and distinguish between boolean and non-boolean features to provide a deeper understanding of feature models of consensus algorithms. In Sect. 4, we discuss how the feature model contributes to the state of the art around consensus algorithms, and we argue that, while consensus algorithms are important, their selection generally fully depends on the DLT platform that is selected first. We conclude and summarize our study in Sect. 5.

2 Research Approach

This study's main research question is, *"How can knowledge be captured regarding consensus algorithms to support web3 development companies with evaluating, designing, and implementing consensus algorithms?"*. It addresses the challenge of selecting an appropriate consensus algorithm for a distributed ledger technology (DLT). This is due to a large number of available alternatives, each with a wide range of features, and the inherent trade-offs between security, scalability, and decentralization, known as the consensus algorithm trilemma. The research project combines multiple research methods, including a literature study and expert interviews, to create an artifact that supports web3 development companies in evaluating, designing, and implementing consensus algorithms. The literature study identifies the role of DLT and consensus algorithms, extracting alternative consensus algorithms and their features and extracting feature models for consensus algorithms. The expert interviews aim to gather data and evaluate the completeness and usefulness of the preliminary design of the artifact, which will be evaluated in case studies. Finally, the research project uses Myers and Newman guidelines to conduct a series of qualitative semi-structured interviews with experts selected based on their expertise and experience. Table 1 shows the experts participating in this research. Seven domain experts, including Blockchain developers and Consensus algorithms experts from different organizations, have participated in the research to assist us with answering the research questions. Before reaching out to potential domain experts, a role description was created to accurately identify their areas of expertise and ensure that the right target group was approached. Subsequently, we sent emails to the chosen experts, providing them with the role description and details regarding our research topic. It is important to note that the selection of experts was carried out in a pragmatic and convenient manner, based on the expertise and experience they had indicated on their LinkedIn profiles. We employed a set of evaluation criteria, such as “Years of experience”, “Expertise”, “Skills”, “Education”, and “Level of expertise”, to guide the selection process. The semi-structured interviews were conducted with experts, and each interview had a duration of 45 to 60 min. To minimize any preconceived notions, we employed a set of open-ended questions to extract as much information as possible from the experts. The interviews were conducted virtually using platforms like Skype and Zoom. Prior consent was obtained from the interviewees to record the interviews, which were later transcribed for analysis. The knowledge obtained from each interview was regularly shared and validated in subsequent interviews to ensure the incremental acquisition of accurate information. Finally, our findings and interpretations were presented to the interview participants for their final approval.

Table 1. The interview participants were experts in consensus algorithm design. Due to the specialized nature of this expertise, the response rates were low, but the quality of the interviews was high.

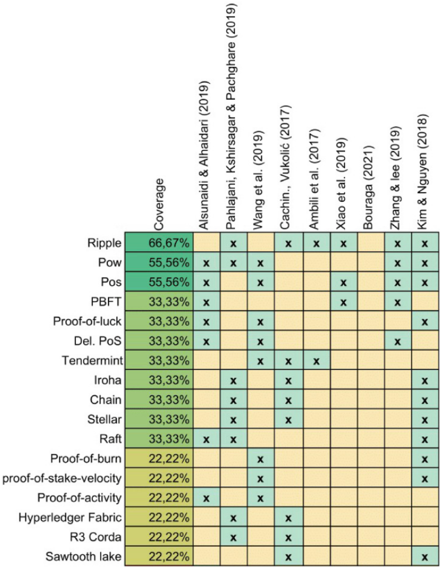
Occupation	Company	Years of Experience
Co-Founder	Lisk	5
Consensus Researcher	Humanode	6
Blockchain developer	Gimly Blockchain projects	4
Blockchain developer	dappdevelopment.com	6
Founder	Emerging Horizons	3
Co-Founder	WBNoDe	8
Consensus algorithm developer	Hyperledger Fabric	4

Concensus Algorithms - The literature on benchmarking the consensus algorithms for blockchains includes several studies, such as [7,11,13]. While two of these studies propose a Boolean decision tree, they have limitations, such as a restricted set of alternatives and features. The survey presented by Fu et al. [13] lacks robustness as it offers limited features and alternatives. So while these studies laid an excellent foundation for this study, we decided to dive deeper into the features that consensus algorithms provide and evaluate these with practitioners to provide an actionable set of knowledge about the features of consensus algorithms. Researchers, consensus algorithm designers, and consensus algorithm implementers can form better technology selection decisions with such knowledge.

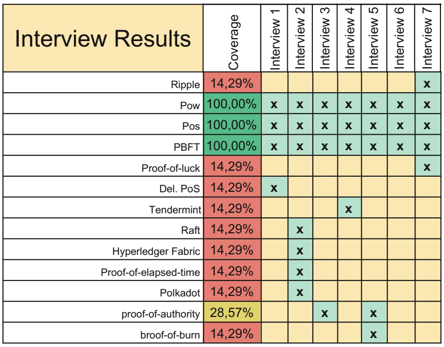
Based on the literature study, we collected the different consensus algorithms as alternatives and their features that define a consensus algorithm. These features and alternatives are required for consensus algorithm selection. This wide variety of features and different algorithms has led to classifying consensus algorithm selection as a Multi-Criteria Decision-Making (MCDM) problem [5]. The full overview of all consensus algorithms and their sources can be found in Table 2.
From the consensus algorithms identified in the literature, many were considered by the interviewees to be either unused or unfamiliar to them. The responses regarding the number of significant alternatives can be found in Table 2.

The selection of consensus algorithms is closely linked to the choice of distributed ledger platforms, which greatly influences the type of consensus algorithms considered. As a result, lesser-known algorithms, such as proof-of-play, which are not currently employed by any major platforms, are generally not considered due to a lack of trust in their reliability.

Table 2. These tables compare some of the consensus algorithms mentioned in the literature (left) to those confirmed as relevant in interviews (right).



	Coverage	Alsunaidi & Alhaidari (2019)	Pahlajani, Kshirsagar & Pachghare (2019)	Wang et al. (2019)	Cachin., Vukolić (2017)	Ambili et al. (2017)	Xiao et al. (2019)	Bouraga (2021)	Zhang & lee (2019)	Kim & Nguyen (2018)
Ripple	66,67%		x		x	x	x		x	x
Pow	55,56%	x	x	x					x	x
Pos	55,56%	x		x			x		x	x
PBFT	33,33%	x					x		x	
Proof-of-luck	33,33%	x		x						x
Del. PoS	33,33%	x		x					x	
Tendermint	33,33%			x	x	x				
Iroha	33,33%		x		x					x
Chain	33,33%		x		x					x
Stellar	33,33%		x		x					x
Raft	33,33%	x	x							x
Proof-of-burn	22,22%			x						x
proof-of-stake-velocity	22,22%			x						x
Proof-of-activity	22,22%	x		x						
Hyperledger Fabric	22,22%		x		x					
R3 Corda	22,22%		x		x					
Sawtooth lake	22,22%				x					x

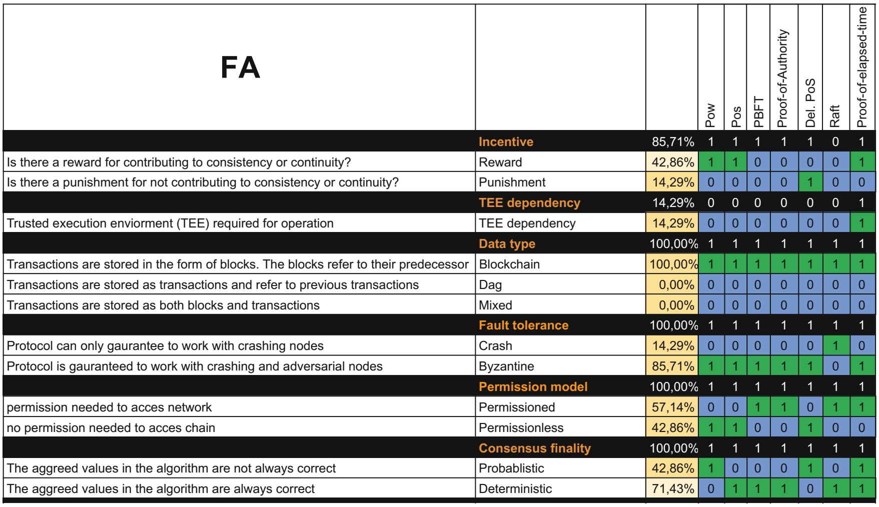


Interview Results	Coverage	Interview 1	Interview 2	Interview 3	Interview 4	Interview 5	Interview 6	Interview 7
Ripple	14,29%							x
Pow	100,00%	x	x	x	x	x	x	x
Pos	100,00%	x	x	x	x	x	x	x
PBFT	100,00%	x	x	x	x	x	x	x
Proof-of-luck	14,29%							x
Del. PoS	14,29%	x						
Tendermint	14,29%				x			
Raft	14,29%		x					
Hyperledger Fabric	14,29%		x					
Proof-of-elapsed-time	14,29%		x					
Polkadot	14,29%		x					
proof-of-authority	28,57%			x		x		
broof-of-burn	14,29%					x		

3 Feature Model

We have used data from domain experts and literature studies for identifying consensus algorithms. Each feature is assigned a Boolean or non-Boolean data type. Consensus algorithm Boolean features fall into three categories: design, security, and performance, with trade-offs among them. PBFT algorithms have higher throughput than PoW algorithms [13].

Table 3. This table displays Boolean Features, consensus algorithms, and mapping. 1s indicate supported consensus algorithm features while 0s signify a lack of support or insufficient evidence based on documentation analysis [6].



FA			Pow	Pos	PBFT	Proof-of-Authority	Del. PoS	Raft	Proof-of-elapsed-time
	Incentive	85,71%	1	1	1	1	1	0	1
Is there a reward for contributing to consistency or continuity?	Reward	42,86%	1	1	0	0	0	0	1
Is there a punishment for not contributing to consistency or continuity?	Punishment	14,29%	0	0	0	0	1	0	0
	TEE dependency	14,29%	0	0	0	0	0	0	1
Trusted execution enviorment (TEE) required for operation	TEE dependency	14,29%	0	0	0	0	0	0	1
	Data type	100,00%	1	1	1	1	1	1	1
Transactions are stored in the form of blocks. The blocks refer to their predecessor	Blockchain	100,00%	1	1	1	1	1	1	1
Transactions are stored as transactions and refer to previous transactions	Dag	0,00%	0	0	0	0	0	0	0
Transactions are stored as both blocks and transactions	Mixed	0,00%	0	0	0	0	0	0	0
	Fault tolerance	100,00%	1	1	1	1	1	1	1
Protocol can only gaurantee to work with crashing nodes	Crash	14,29%	0	0	0	0	0	1	0
Protocol is gauranteed to work with crashing and adversarial nodes	Byzantine	85,71%	1	1	1	1	1	0	1
	Permission model	100,00%	1	1	1	1	1	1	1
permission needed to acces network	Permissioned	57,14%	0	0	1	1	0	1	1
no permission needed to acces chain	Permissionless	42,86%	1	1	0	0	1	0	0
	Consensus finality	100,00%	1	1	1	1	1	1	1
The aggreed values in the algorithm are not always correct	Probablistic	42,86%	1	0	0	0	1	0	1
The aggreed values in the algorithm are always correct	Deterministic	71,43%	0	1	1	1	0	1	1

Design features of consensus algorithms refer to the structure and framework of the algorithm and include incentives, consensus finality, candidate formation, candidate configuration, leader selection, committee formation, and committee configuration. Incentives are the motivations for participating nodes to engage in the mining process and contribute to the consensus algorithm. These incentives can include rewards, such as cryptocurrency tokens, to encourage nodes to perform computational work and secure the network [19]. Consensus finality refers to the level of certainty or irreversibility that the consensus algorithm can achieve [21]. It determines when a transaction or block is considered finalized and cannot be altered or reversed. Candidate formation pertains to the criteria that nodes must meet to be eligible to participate in the consensus-building process. This includes factors such as node reputation, stakeholding, or computational power. Candidate configuration relates to how the group of participating nodes evolves during the consensus-building process. It involves determining which nodes are eligible to become candidates for adding new entries to the blockchain. Leader selection involves the process of choosing a node or a group of nodes responsible for proposing and validating new blocks or transactions [18]. This mechanism can vary depending on the consensus algorithm, with different approaches such as round-robin selection, random selection, or election-based selection. Committee formation refers to the process of selecting a subset of nodes responsible for validating and verifying new entries in the blockchain. These committees ensure the accuracy and integrity of the consensus process. Committee configuration focuses on how the composition of the committee evolves over time. This may include adding or removing nodes based on certain criteria or adjusting the committee size for improved scalability [9].

We conducted qualitative semi-structured interviews with experts to explore their knowledge regarding consensus algorithm features. The design features of consensus algorithms, except for two new features, Trusted Execution Environment (TEE) and file structure, have remained unchanged [3]. TEE relates to how proof is obtained to mine a block, while file structure pertains to the inherent structure of the consensus algorithm and is critical in determining leader selection, candidate formation, and committee formation.

Performance Features - This category of features focuses on how efficiently a consensus algorithm operates. The most important features in this category are throughput and latency. Throughput measures the number of new data entries a consensus algorithm can process per second, while latency measures the time for any data entry to be verified. Scalability is another characteristic in this category, indicating whether a consensus algorithm can function effectively when faced with many transactions or nodes. The extent to which a consensus algorithm can scale differs significantly among different algorithms. Lastly, fault tolerance is a feature that describes whether a consensus algorithm can tolerate Byzantine or crash faults.

One interviewee proposed a feature regarding the upgradability of consensus, stating that in some systems, a set of transactions can change the whole system and synchronize the entire network independently with the consensus at every node. The same developer referred to Substrate as a network that employs such upgradability methods. Another expert emphasized that sustainability is critical for companies building a ledger. They stated that some parties would not build their application on a proof-of-work network as it is not sustainable enough. This concern has a significant impact on the decision-making process, as developers with sustainability in mind tend to choose an algorithm other than proof-of-work, given the significant amount of energy it requires to operate [14].

Security Features - The security features of a consensus algorithm pertain to the degree of protection it provides against various attack vectors and threats. The primary attack vectors include a 51% attack, a Sybil attack, a denial of service (DoS) attack, and an eclipse attack. A 51% attack occurs when a malicious entity controls more than 50% of the network's computing power and can modify the ledger according to their will [4]. On the other hand, a Sybil attack is an attack in which the attacker creates fake identities to gain influence within the network. In a DoS attack, the attacker disrupts the ledger service by making it unavailable to other nodes. Lastly, an eclipse attack is a type of attack where the attacker takes over other nodes and forces them to only communicate with other malicious nodes. During our interviews, we observed that most participants were not familiar with the eclipse attack, and even those who were aware of it did not consider it significant. As a result, we did not include it in the feature model. Additional security features include authentication, non-repudiation, censorship resistance, and adversarial tolerance. Authentication requires nodes to authenticate themselves before participating in consensus. Censorship resistance ensures that no one can censor the data transmitted across the network [11]. Non-repudiation guarantees that no one can deny making a data entry in the ledger. Adversarial tolerance indicates the maximum percentage of adversarial nodes that a consensus algorithm can withstand (Fig. 1).

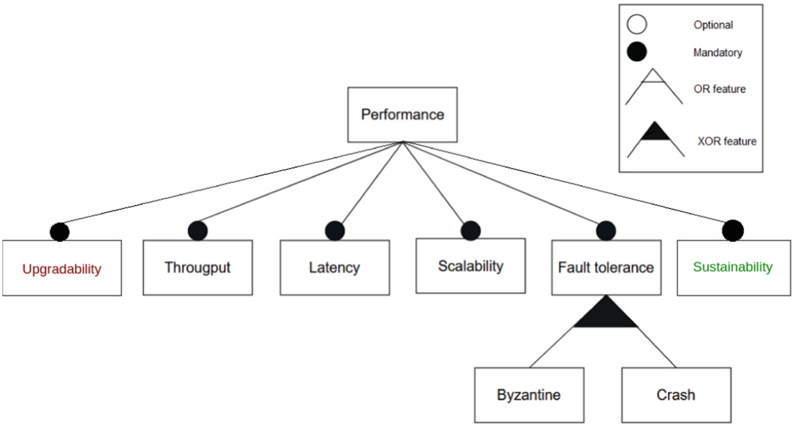


Fig. 1. Feature model of consensus algorithm performance features. Features that were proposed in the interviews but did not find support in literature or interviews are colored in red, whereas features that were added during the expert interviews with support in literature are shown in green.

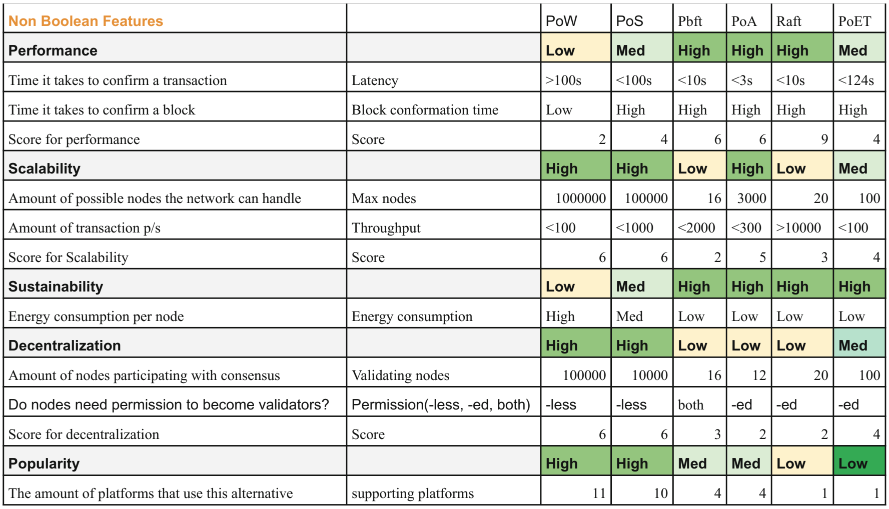
Non-Boolean Algorithm Features - The experts identified five non-Boolean consensus algorithm features, being “Popularity in the market”, “Maturity of the company”, “Developer Resources (People)”, “Sustainability”, and “Scalability”. The assigned values for these features are based on a 3-point Likert scale (High, Medium, and Low) and are used to evaluate a specific consensus algorithm.

Features of Consensus Algorithms - Data was extracted from various sources, including web pages, white papers, scientific papers, documentation, forum discussions, books, videos, and dissertations, to develop the initial list of consensus algorithm features. The initial list comprised 43 Boolean and five non-Boolean features. Subsequently, seven domain experts were involved in the research phase to refine the list of potential consensus algorithm features [7].

The Boolean and non-Boolean feature alternative mappings can be found in Table 3 and Table 4, respectively [6]. The former table contains binary codes for each Boolean feature indicating whether it is present. The yellow column shows the percentage of alternative algorithms with a particular feature. Table 4 assigns a score to each non-Boolean feature based on parameters such as transaction latency, block confirmation time, transaction throughput, the maximum number of nodes, energy consumption, number of validating nodes, permission model, and popularity. The parameters were carefully selected to ensure their relevance and accuracy in evaluating the features. Specifically, the performance score was calculated using transaction latency and block confirmation time as parameters. Transaction latency is the time taken to confirm a single transaction and is measured in seconds. The maximum time to ensure a transaction is used to calculate the score for this parameter. Block confirmation time, on the other hand, is the time required to confirm a single block and is categorized into three discrete values: “low”, “middle”, and “high”. These values are used in the calculation of the performance score. The consensus algorithms’ scalability was calculated using transaction throughput and the maximum number of nodes. Transaction throughput represents the number of transactions the network can process per second and is expressed in maximum transactions per second. The maximum number of nodes is the maximum number of nodes that a consensus algorithm can accommodate before its performance is significantly impacted. This parameter is measured in terms of the total number of nodes. To measure sustainability, energy consumption was used as a parameter. Energy consumption refers to the amount of energy consumed by a ledger of a given alternative type and is categorized into three discrete values: “low”, “middle”, and “high”. Decentralization is measured using two parameters: the number of validating nodes in the network and the permission model. The number of validating nodes represents the total number of nodes that can participate in the consensus-finding process and is expressed in the number of validating nodes. The permission model determines whether a consensus algorithm operates in a permissioned or permissionless network. In a permissioned network, validating nodes require permission to participate in the consensus process, whereas in a permissionless network, there are no such restrictions. The popularity of a consensus algorithm is determined by the number of platforms that use it [4].

Conflicting views on attack vectors were resolved by prioritizing studies that provide clear reasoning for their conclusions. These algorithms are equipped with a mechanism that safeguards against particular attacks. This information has been identified in several surveys that have explored various consensus algorithms. However, some studies have reached different conclusions regarding attack vectors. For instance, [2] and [4] diverge views on whether proof-of-stake defends against double-spending attacks. In cases where conflicting information exists, studies that provide clear reasoning for their conclusions on attack vectors are preferred over those that assert that a particular algorithm offers protection against a given attack.

Table 4. The mapping among non-Boolean features and selected alternatives.



Non Boolean Features		PoW	PoS	Pbft	PoA	Raft	PoET
Performance		**Low**	**Med**	**High**	**High**	**High**	**Med**
Time it takes to confirm a transaction	Latency	>100s	<100s	<10s	<3s	<10s	<124s
Time it takes to confirm a block	Block conformation time	Low	High	High	High	High	High
Score for performance	Score	2	4	6	6	9	4
Scalability		**High**	**High**	**Low**	**High**	**Low**	**Med**
Amount of possible nodes the network can handle	Max nodes	1000000	100000	16	3000	20	100
Amount of transaction p/s	Throughput	<100	<1000	<2000	<300	>10000	<100
Score for Scalability	Score	6	6	2	5	3	4
Sustainability		**Low**	**Med**	**High**	**High**	**High**	**High**
Energy consumption per node	Energy consumption	High	Med	Low	Low	Low	Low
Decentralization		**High**	**High**	**Low**	**Low**	**Low**	**Med**
Amount of nodes participating with consensus	Validating nodes	100000	10000	16	12	20	100
Do nodes need permission to become validators?	Permission(-less, -ed, both)	-less	-less	both	-ed	-ed	-ed
Score for decentralization	Score	6	6	3	2	2	4
Popularity		**High**	**High**	**Med**	**Med**	**Low**	**Low**
The amount of platforms that use this alternative	supporting platforms	11	10	4	4	1	1

4 Discussion

Blockchain consensus algorithm selection is a crucial aspect that requires careful consideration, as the technology's affordances and consequences have long-term effects on the blockchain application. Changing consensus algorithms can be costly and complex, as evident in Ethereum's transition to a proof-of-stake consensus algorithm [10]. Hence, selecting an algorithm that best fits the current and future requirements of the blockchain project is essential. Several decision-support models have been proposed for selecting the appropriate consensus algorithm [7, 12], and modular blockchain platforms have become popular due to their flexibility in selecting consensus algorithms [1].

In this research project, we interviewed engineers and consensus algorithm designers, which introduces a validity concern. While engineers focus on the consequences of selecting a platform based on the consensus algorithm, algorithm designers are more concerned with the principles of the algorithm itself. Designers have more freedom to consider the adaptability of the algorithm, while engineers typically work within the given framework. These interviews revealed that consensus algorithm selection is not a significant concern for most distributed ledger application developers during the building process. This significantly impacts the natural validity of the research project. Initially, our search focused on consensus algorithms and platform developers, but finding and convincing them to participate was challenging. This difficulty can be attributed to the need for more highly specialized employees in companies located primarily in the United States. As a result, we expanded our search to include blockchain application developers and consultants. While this introduces the previously mentioned validity concern, these additional experts provided valuable insights. They had extensive experience with various blockchain application requirements, which often influenced the choice of consensus algorithm features. Consequently, we could still deduce the essential features in consensus algorithm selection. Another validity concern emerged during the interviews when experts proposed features related to platform selection rather than consensus algorithm selection. This indicated confusion among some experts regarding which features are attributed to consensus algorithms and which belong to other layers of a blockchain application. We conducted a second literature review after the interviews to address this concern. Each proposed feature was carefully examined to determine its relevance to the algorithm layer or other layers of the blockchain application. As a result, two features suggested by interviewees were excluded.

5 Conclusion

In this work, we outline the features that distributed ledger technology designers need to consider with regard to consensus algorithms. While these decisions are not made regularly, we show they have significant consequences for the platform.

We find consensus algorithms are not frequently evaluated, designed, and implemented. While many others have offered decision support systems for consensus algorithm selection, we propose, as our scientific contribution, that more in-depth analysis and reporting of consensus algorithms is necessary. Subsequently, we use the feature modeling language to elicit the relevant features for consensus algorithms to provide a complete overview of what consensus algorithms have to offer than was available in the literature. The feature models in Sect. 3 are useful for scientists working in consensus algorithms and practitioners evaluating, designing, and implementing consensus algorithms.

Our practical contribution is the list of consensus algorithms that are supplied in Tables 2, and the features identified in some of the more common consensus algorithms in Tables 3, and 4. Experts evaluated and verified these tables to ensure our data was complete and correct. Using our conceptual model, we unearthed six of the most common consensus algorithms and mapped 48 distinguishing, unique features of these algorithms for use by industry practitioners. Practitioners designing distributed ledger technologies can use these models in the future to understand the trade-offs of the architectural decisions they will be making.

The findings of this research project open up several potential research for future work. To enhance the precision of the decision model, it is crucial to conduct a dedicated performance study that comprehensively compares the performances of a wide range of consensus algorithms. Currently, there is a scarcity of published studies comparing consensus algorithm performance, which can be attributed to the challenges associated with comparing performance due to the multitude of factors involved that extend beyond consensus alone. Existing studies that do compare performances often focus on a limited subset of algorithms in diverse environments. Although a combination of these studies has allowed us to deduce performance differences between algorithms, a dedicated performance study would offer a more precise and accurate assessment of the performance characteristics of each consensus algorithm. Such a study would provide valuable insights for informed decision-making in selecting the most suitable consensus algorithm for specific distributed ledger applications.

References

1. Alchemy: Modular vs. monolithic blockchains (2022). Accessed 5 Oct 2022
2. Alsunaidi, S.J., Alhaidari, F.A.: A survey of consensus algorithms for blockchain technology. In: 2019 International Conference on Computer and Information Sciences (ICCIS), pp. 1–6. IEEE (2019)
3. Ampel, B., Patton, M., Chen, H.: Performance modeling of hyperledger sawtooth blockchain. In: 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 59–61. IEEE (2019)
4. Bamakan, M.H., Motavali, A., Bondarti, A.B.: A survey of blockchain consensus algorithms performance evaluation criteria. Expert Syst. Appl. **154**, 113385 (2020)
5. Baninemeh, E., Farshidi, S., Jansen, S.: A decision model for decentralized autonomous organization platform selection: three industry case studies. Blockchain: Res. Appl. **4**, 100127 (2023)
6. Baninemeh, E., Jansen, S., Pronk, B.: A feature model of consensus algorithms in distributed ledger technology. https://bit.ly/42TYrb8
7. Bouraga, S.: A taxonomy of blockchain consensus protocols: a survey and classification framework. Expert Syst. Appl. **168**, 114384 (2021)
8. Cachin, C., Vukolić, M.: Blockchain consensus protocols in the wild. In: 31 International Symposium on Distributed Computing (2017)
9. Chaudhry, N., Yousaf, M.M.: Consensus algorithms in blockchain: comparative analysis, challenges and opportunities. In: 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), pp. 54–63. IEEE (2018)
10. Farshidi, S., Jansen, S., España, S., Verkleij, J.: Decision support for blockchain platform selection: three industry case studies. IEEE Trans. Eng. Manage. **67**(4), 1109–1128 (2020)
11. Ferdous, M.S., Chowdhury, M.J.M., Hoque, M.A., Colman, A.: Blockchain consensuses algorithms: a survey. arXiv preprint arXiv:2001.07091 (2020)
12. Filatovas, E., Marcozzi, M., Paulavičius, R.: A MCDM-based framework for blockchain consensus protocol selection. Expert Syst. Appl. **204**, 117609 (2022)
13. Fu, X., Wang, H., Shi, P.: A survey of blockchain consensus algorithms: mechanism, design and applications. Sci. China IS **64**(2), 1–15 (2021)
14. Jones, B.A., Goodkind, A.L., Berrens, R.P.: Economic estimation of bitcoin mining's climate damages demonstrates closer resemblance to digital crude than digital gold. Sci. Rep. **12**(1), 1–10 (2022)
15. Kannengiesserer, N., Lins, S., Dehling, T., Sunyaev, A.: Trade-offs between distributed ledger technology characteristics. ACM Comput. Surv. (CSUR) **53**(2), 1–37 (2020)
16. Monrat, A.A., Schelén, O., Andersson, K.: A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access **7**, 134–151 (2019)
17. Ølnes, S., Ubacht, J., Janssen, M.: Blockchain in government: benefits and implications of distributed ledger technology for information sharing (2017)

18. Ongaro, D., Ousterhout, J.: In search of an understandable consensus algorithm. In: 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 2014), pp. 305–319 (2014)
19. Singh, A., Kumar, G., Saha, R., Conti, M., Alazab, M., Thomas, R.: A survey and taxonomy of consensus protocols for blockchains. J. Syst. Archit. **127**, 102503 (2022)
20. Suciu, G., Nădrag, C., Istrate, C., Vulpe, A., Ditu, M.C., Subea, O.: Comparative analysis of distributed ledger technologies. In: 2018 Global Wireless Summit (GWS), pp. 370–373. IEEE (2018)
21. Yadav, A.K., Singh, K., Amin, A.H., Almutairi, L., Alsenani, T.R., Ahmadian, A.: A comparative study on consensus mechanism with security threats and future scopes: blockchain. Comput. Commun. **201**, 102–115 (2023)