Expert Systems With Applications 154 (2020) 113385

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

A survey of blockchain consensus algorithms performance evaluation criteria

Seyed Mojtaba Hosseini Bamakan[a,b,*], Amirhossein Motavali[a], Alireza Babaei Bondarti[a]

[a] Data Science Research Center, Yazd University, Yazd, 89195-741, Iran
[b] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China

ARTICLE INFO

Article history:
Received 7 November 2019
Revised 18 February 2020
Accepted 12 March 2020
Available online 13 April 2020

Keywords:
Blockchain
Consensus algorithms
Performance evaluation criteria
Security vulnerability
Trust and permission models

ABSTRACT

How to reach an agreement in a blockchain network is a complex and important task that is defined as a consensus problem and has wide applications in reality including distributed computing, load balancing, and transaction validation in blockchains. Over recent years, many studies have been done to cope with this problem. In this paper, a comparative and analytical review on the state-of-the-art blockchain consensus algorithms is presented to enlighten the strengths and constraints of each algorithm. Based on their inherent specifications, each algorithm has a different domain of applicability that yields to propose several performance criteria for the evaluation of these algorithms. To overview and provide a basis of comparison for further work in the field, a set of incommensurable and conflicting performance evaluation criteria is identified and weighted by the pairwise comparison method. These criteria are classified into four categories including algorithms' throughput, the profitability of mining, degree of decentralization and consensus algorithms vulnerabilities and security issues. Based on the proposed framework, the pros and cons of consensus algorithms are systematically analyzed and compared in order to provide a deep understanding of the existing research challenges and clarify the future study directions.

Contents

* Corresponding author.
E-mail addresses: smhosseini@outlook.com, smhosseini@yazd.ac.ir (S.M.H. Bamakan).

https://doi.org/10.1016/j.eswa.2020.113385
0957-4174/

1. Introduction

Undoubtedly, blockchain technology is one of the largest technologies with an enlightening future. Although the first application of blockchain technology was Bitcoin as a cryptocurrency, other applications of this technology have also gained attention from governmental and industrial sectors. According to the World Economic Forum survey (Forum, 2015), by 2027, blockchain will store ten percent of global GDP.

Blockchain technology was introduced for the first time by a group of researchers (Haber & Stornetta, 1990) and until the establishment of Bitcoin by Satoshi Nakamoto in 2008 (Nakamoto, 2019), it did not have commonplace applications. However, it should be noted that in the recent years it has been used in different domains such as biomedical (Drosatos & Kaldoudi, 2019), supply chain management (Kshetri, 2018; Min, 2019), and registering smarts contracts (Macrinici, Cartofeanu, & Gao, 2018). A systematic survey on the application of blockchain technology in various fields is presented in (Casino, Dasaklis, & Patsakis, 2019). Blockchain as a distributed and decentralized database is a sequence of blocks that in each block a list of transactions is accumulated. Each block has three major sections: data, hash block, and previous hash block. Hash determines the identity of each block like a fingerprint and is unique for each block. The information of each block is indicated by Hash. When a transaction is registered in a block, its hash number is calculated in an encryption block containing information and is obtained by mathematical rules. Each block contains the hash of the previous block; thus the blocks are connected to each other. Any changes made in the information of a block cause changes in its hash number. Therefore, any illegal changes to the information of the blocks can change its hash number and this will make the block become invalid for the next blocks (Zheng, Xie, Dai, Chen, & Wang, 2017a). Fig. 1, illustrates the structure of bitcoin blockchain for three blocks.

As presented in Fig. 1, the first block is called the Genesis block and because there is no other block before it, the previous hash amount is equal to zero. Each block can contain thousands of transaction records that are coded by a hash function before broadcasting to the network. Blockchain uses Merkle tree function to generate a final hash value as a hash pointer (hash of current block) and each block contains the hashcode of the previous block to preserve the connectivity of blocks. Merkle tree is a hash tree-like data structure that stores the transactions in a binary tree format. Each leaf node of the tree stores the hash value of transactions and a non-leaf node contain the hash of the hashes of its two corresponding child nodes and finally, the root of this tree called Merkle digest/root. Using a Merkle tree function will reduce the cost of data transmission and computing resources (Panarello, Tapas, Merlino, Longo, & Puliafito, 2018). In summary, the process of mining or validating a new block by the proof of work algorithm is required to do an exhaustively querying a cryptographic hash function to find a nonce in such a way that satisfies a predefined condition. Let's suppose $H()$ as a hash function and x as a Merkle root of the transactions in a block. Hence, Wang et al. (2018a) defined the PoW puzzle and its solution as follows:

> "Given an adjustable hardness condition parameter h, the process of PoW puzzle solution aims to search for a solution string, **nonce**, such that for a given string x assembled based on the candidate block data, the hashcode of the concatenation of x and nonce is smaller than a target value $\boldsymbol{D}(\boldsymbol{h})$"

$$Target\ hash = H(x\,nonce) \leq D(h)$$

Where *nonce* abbreviation for "number only used once" defined as a number added to a hashed block in a blockchain to meet the predefined difficulty level and for some fixed length of bits L, $D(h) = 2^{L-h}$.

A general classification divides blockchain into three categories including public blockchain, consortium blockchain and private blockchain (Korpela, Hallikas, & Dahlberg, 2017):

- ***Public blockchains*** are considered as a sort of decentralized permissionless blockchain in which information is presentable for all network members and all can participate in its acceptance. We can address Bitcoin and Ethereum as examples of a public blockchain. This type of blockchain is secure because of its consensus mechanism that achieves agreement among all peers. These consensus algorithms include proof of work (PoW) and proof of stake (PoS), etc.
- ***Consortium blockchains*** are also known as federated blockchains in which the information is presentable for all people, but its change and acceptance is only possible for determined groups; For example, the presentation of marketing products by blockchain. Consortium blockchains are mostly used in the banking sector (Dib, Brousmiche, Durand, Thea, & Hamida, 2018). Here, the idea is to distribute the power among a number of authorities instead of having a single full control authority to make a collective and unbiased decision. R3 (Banks), EWF (Energy) and B3i (Insurance) are some examples of consortium blockchains.

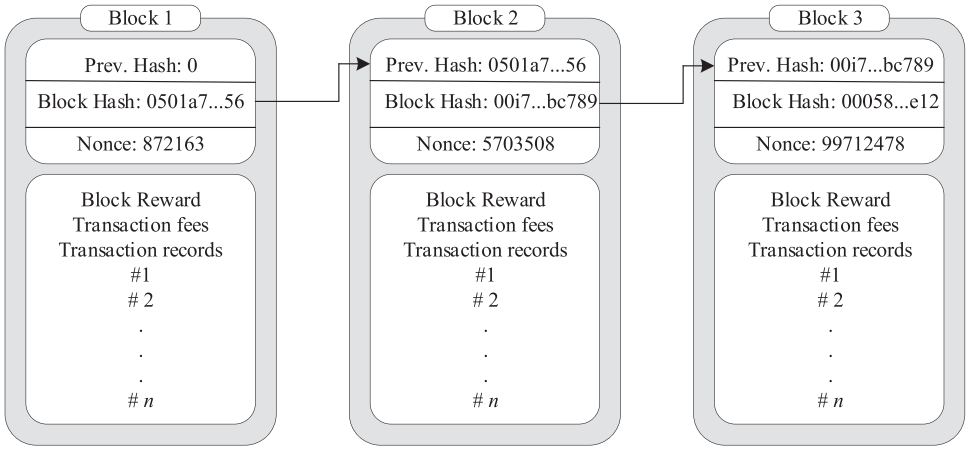

Fig. 1. The bitcoin blockchain structure.

- ***Private blockchains*** are permissioned blockchains in which the information is presentable for a special group and its change acceptance is only possible by an authorized group, e.g. payroll system by blockchain. This is a centralized blockchain that there is a central authority that determines the permission on who can read, write or participate in the blockchain. Hence the consensus mechanism in private blockchains is defined by a single central authority.

These three kinds of blockchain have differences based on how to reach consensus among the participates. For example, in public blockchain all miners determine the consensus, however, in consortium blockchain and private blockchain, consensus determination can be done by a selected set of nodes or one organization, respectively.

The security and validation of a block is an important task that is reached by a certain mechanism called consensus algorithms. In the distributed systems, how attaining a consensus among the unreliable nodes has been considered as a Byzantine Generals Problem (BGP((Leslie Lamport, Shostak, & Pease, 1982) that a group of army generals has surrounded a city. Some generals prefer to attack the city and others prefer to retreat. Attacking the city by some of the generals will fail. Therefore, they should agree on attack or retreat. Reaching consensus in a distributed environment is a challenge. For blockchain which has a distributed system, the consensus is a challenge, too (Zheng et al., 2017a). Blockchain is a decentralized network that is there is not a central node to observe and check all transactions. Thus, there is a need to design protocols that indicate all transactions are valid. So, the consensus algorithms are considered as the core of each blockchain. In the distributed systems the consensus has become a problem in which all network members (nodes) agree on accept or reject of a block. When the new block is accepted by all network members, it can append to the previous block.

As mentioned, the main concern in the blockchains is how to reach consensus among network members. A wide range of consensus algorithms has been designed in which each of them has a lot of cons and pros. This amount of existing consensus algorithms could cause confusion in their selection and apply to cope with real-world problems. In this paper, the main objective is to provide a comparative platform to evaluate the performance of the most well-known blockchain consensus algorithms in terms of identified technical and economic criteria that in some cases are non-commensurable and conflicting criteria. Hence, there is a need for identifying a set of complete performance evaluation criteria that cover all aspects of consensus algorithms, besides deep understanding the constraints in the existing algorithms to reach consensus among the peers and ensure the security of information in the blockchain. In the following, the main contributions of this paper are mentioned.

- Presenting a comprehensive and comparative review of the state-of-the-art blockchain consensus algorithms.
- Identifying and weighting the most applicable and effective criteria as an analytical framework to evaluate the pros and cons of consensus mechanisms.
- Discussing the existing research challenges, open issues and new directions to enlighten future research.

The remainder of the paper is organized as follows. In Section 2, we review the related work in this domain. In Section 3, we present an overview of the most well-known blockchain consensus algorithms. An analytical framework includes four categories of criteria and their sub-categories to evaluate consensus algorithms are presented in Section 4. In Section 5, we discuss the challenges and limitations of these algorithms and suggestions for future research. Finally, we conclude this paper in section 6.

2. Related work

In this section, we aim to identify the literature related to consensus algorithms in general and also the related studies on comparing the consensus algorithms. Hence, our methodology consists of a systematic search in the journal and conference papers as well as books and articles to notice criteria and metrics that researchers used to compare consensus algorithms. To achieve this, we first identified the key search terms including but not limited to “consensus algorithm, comparative analysis, blockchain technology, distributed ledger, PoW, Proof-of-work, PoS” and more. In addition to these keywords, we also looked for alternative keywords and synonyms with adding AND/OR Boolean expressions to explore more related articles in various academic databases such as Science Direct, IEEE Xplore, SpringerLink, and ACM digital library. In the next step, we assessed the quality of the papers and omitted the ones that were not within the scope of our survey.

The ability to reach agreement in a board of “Partial synchronous” systems was first issued in 1988 by Lynch, Dwork, and Stockmeyer (Dwork, Lynch, & Stockmeyer, 1988). Oki and Liskov in 1988 also published a protocol used to reach an agreement in “view stamped replication” for a distributed transaction (Oki & Liskov, 1988). A particularly elegant formalism protocol named Paxos was then introduced with strong similarities to the Oki and Liskov protocol (Leslie Lamport, 1998).

Paxos is the first comprehensive family of protocols introduced for reaching consensus in a network with faulty or unreliable processes (Turner, 2007). Solving the consensus problem is especially

difficult when a participant is dishonest, or the communications between the participants experience failures (Pease, Shostak, & Lamport, 1980). The Paxos protocol was first submitted in 1990 and then published as a journal article titled "The Part-Time Parliament" by Leslie Lamport in 1998 (Leslie Lamport, 1998). The authors in (Fischer, Lynch, & Paterson, 1985) proved that no deterministic consensus protocol would guarantee process in an asynchronous network. However, Paxos can guarantee the consistency and safety of the network. According to (Leslie Lamport, 1998), although Paxos is difficult to understand and hard to implement, it is the current standard for both teaching and implementing consensus algorithms (Ongaro & Ousterhout, 2014). Besides talking about the Paxos protocol, Ongaro also compared Paxos with another consensus algorithm named "Raft". Raft is not only much easier to understand and implement, but also has no performance penalty and has several open-source implementations.

Consensus algorithms have many applications, such as deciding on the validity of distributed transactions in cryptocurrencies like bitcoin, confirming the identity of the leader for a distributed task, ensuring the consistency among state machine replicas, and synchronizing them. Google has a distributed lock service named Chubby that uses the Paxos algorithm to keep its replica consistent in case of a failure (Burrows, 2006). A consensus algorithm can be used for solving real-world problems like time synchronization among distributed systems (Schenato & Gamba, 2007), load-balancing in networks, ranking webpages and controlling multiple UAVs, robots and agents. Gulzar et al. (Gulzar, Rizvi, Javed, Munir, & Asif, 2018) describe the current status and future of cooperative control consensus in multi-agent systems while presenting a comparative review on "multi-agent cooperative control consensus".

Exciting intense attention since 2008, blockchain as a type of a distributed ledger in maintaining a tamper-proofed record of transactional data, also relies on consensus algorithms. Blockchain is the basis of cryptocurrencies like Bitcoin, Ethereum, Ripple, Litecoin, NEM, and etc. According to CoinMarketCap, we have currently around 4900 cryptocurrencies and a total market cap of approximately $200 billion circulating in the market today and what makes these cryptocurrencies different from each other is their specific consensus algorithm. "A survey about the consensus algorithms used in blockchain highlights" is a paper published in 2018 by (G.-T. Nguyen & Kim, 2018) which highlights a bunch of 32 different consensus algorithms categorized into two major types including Proof-based and Vote-based algorithms. In this survey, in addition to illustrating the advantages and drawbacks of each type, a comparison between these two types based on some of their prominent characteristics is provided. Some factors such as agreement making basement, decentralization, trust model, security threat, and the number of the nodes executing are used to provide a comparison between the vote-based and proof-based algorithms. Based on this work, the authors in (Alsunaidi & Alhaidari, 2019) considered more criteria and classified them into four major groups including, main features, incentive, scope and exposure likelihood. Although this paper provides a basis for comparison between the algorithms, it has missed some important criteria such as trust model and decentralization levels.

Another comparative analysis of the most famous blockchain consensus algorithms is done by Bach, Mihaljevic, & Zagar, (2018) with the focus on the algorithmic steps taken by each consensus algorithm, the method of rewarding validators and the security risks of the algorithms. Some information related to the power consumption and the number of possible transactions per second of selected cryptocurrencies is also available in (Bach et al., 2018). The criteria used in this study include security, scalability (number of TPSs) and power consumption with the focus on PoW and PoS algorithms. By proving the deficiencies of PoW in terms of power consumption and Scalability, they concluded that "the Proof of Work system, which is by far the most popular consensus algorithm, will eventually be replaced by newer, more efficient algorithms".

More recent surveys on this domain have been reviewed the main properties of the consensus algorithms and analyzed the performance and application scenarios of different consensus mechanisms, besides summarizing the limitations and future development of blockchain technology (Mingxiao, Xiaofeng, Zhe, Xiangwei, & Qijun, 2017a; Sankar, Sindhu, & Sethumadhavan, 2017; Wang et al., 2018b). However, there is a gap in the existing literature that do not provide a complete set of criteria for comparative analytics, hence this paper focuses on identifying the main consensus algorithms performance evaluation criteria and categories them into four groups including algorithms throughput, the profitability of mining, decentralization and consensus algorithms vulnerabilities.

3. Blockchain consensus algorithms

How to reach an agreement in a blockchain network is a complex and significant task. New transaction records would be added to the blockchain since the new block is verified by all nodes in the network. It should be noted that once blocks are verified, it is not feasible to modify or delete them. The structure of blockchains is designed to be valid in a trustless and unreliable network with adversarial users. Various methods are designed and developed as consensus algorithms. The number of these algorithms is increasing day to day according to blockchain development. However, in this section, we will introduce the most important consensus algorithms which are widely used in the blockchain networks, and discuss their advantages and disadvantages.

3.1. Proof of Work (PoW)

The most well-known consensus method is Proof of Work (PoW) which was introduced by Nakomoto (Nakamoto, 2019) and is used in Bitcoin. The Proof of Work has been around for many years as a suitable method for currency cryptography. In this method, the computer does many computations to solve a mathematics puzzle. This puzzle-solving is done through the Hash function. Hash is a random and complex mathematical formula that is used for confirmation of the transactions stored in blocks (Salimitari & Chatterjee, 2018). In brief, each block consists of the previous block's hash value, a history of transactions, nonce and current block hash. A miner, that is the computer trying to solve the hash, will try to find a specific value as a nonce in such a way that the hash value meets a pre-defined condition; for example, finding the nonce that will make the 30-first bits of its hash value zero. By changing these pre-defined conditions, the network can be very scalable and flexible to any condition. In PoW, each node of the network is calculating a hash value of the block header. In other words, to reach consensus in the network, miners try to find hash value equal to or smaller than a certain given value. When one node finds the target value, it would broadcast the block to the whole network and all other nodes should confirm the correctness of the hash value. Hence, if the block is validated, all nodes would append this new block to their own chain (Xu, Wang, & Guo, 2017).

The advantage of the Proof of Work algorithm is its high security and decentralization. However, its main disadvantage is that the function of mining and validating blocks wastes a lot of energy. Moreover, the speed and success rate of this hash function highly depends on the computational abilities of the hardware running the hash. (Zheng et al., 2017a). Moreover, although the complexity of the hash function can be scalable, due to the complexity of solving the hash function, solving this puzzle takes some time and therefore, this algorithm is not suitable for large and fast-growing

networks that require huge numbers of transactions per second (Alsunaidi & Alhaidari, 2019). In summary, its pros consist of having a decentralized structure, high levels of security and acceptable levels of scalability. On the other hand, its cons are less throughput, high block creation time, energy inefficiency, special hardware dependency, high computational cost, and extensive bandwidth requirements.

3.2. Proof of Stake (PoS)

After the Proof of Work, the next common consensus algorithm in blockchain technology is the Proof of Stake. Major problems in proof of work systems, such as energy inefficiency were the reason for creating proof of stake. The PoS algorithm is based on the idea that the creator of the next block should be chosen via various combinations of random selection, his stake supply, and age which can provide good scalability. This idea was introduced in 2011 for Peercoin cryptocurrency and after that was used in others such as Nxt and Blackcoin (Bashir, 2017). Selected node for making the next block, will be chosen through a quasi-random process in which the selection depends on assets stored in the wallet (or pool of shares) relating to that node. This method does not need high computing power for validating any proof and therefore, the miners will receive no reward except for the transaction fees. Although this method does not need proof of work's computing power, it strongly depends on nodes that have the most stake and the blockchain will somehow become centralized. Moreover, there is another common problem for the Proof-of-Stake system called "nothing at stake", which means if a node has nothing in his stake while misbehaving, he is not afraid of losing anything. Therefore, there will be no obstacles for the node to prevent him from misbehaving. A misbehavior example could be making two sets of new blocks for getting double transaction fees (Bach et al., 2018; Salimitari & Chatterjee, 2018).

Leased Proof of Stake is an advanced version of Proof of Stake (PoS) that is used in the Waves Platform (Waves Docs, 2018). In the Leased Proof of Stake, the node that maintains a specific amount of cryptocurrency is eligible to add the next block to the blockchain. However, in the LPoS, on Waves Platform, users can lend a portion of their account into full nodes and receive a percentage as a bonus (Salimitari & Chatterjee, 2018). The higher the amount is lent to a full node, the better the chance for a full node to create a new block. If the full node is selected for building a new block, the lender earns a percentage of the transaction that is collected by the full node later that on (Waves Docs, 2018).

In conclusion, there are some advantages to the PoS based consensus algorithms such as fast block creation time, high throughput, energy efficiency, scalability (but less than PoW) and independence to the special hardware. However, this group of algorithms suffers from form some sort of centralization and lower cost of misbehaving in blockchain networks.

3.3. Delegated Proof of Stake

This algorithm was introduced by Daniel Larimer (Larimer, 2014). This method is an improvement to the Proof of Stake's method so that the nodes select representatives through voting in order to validate blocks (Bashir, 2017). The number of representatives is limited and this will make it possible to organize the network more effectively and each representative can determine the adequate time to publish each block. This method has been used in the Bitshares. However, this limitation on the number of representatives would make the network more centralized. (Salimitari & Chatterjee, 2018). The most important features of this mechanism can be mentioned as scalability, energy-efficiency, and low-cost transactions. Despite all of these benefits, it is a semi-centralized mechanism and is better being used in private blockchains. However, If a selected representative delay or make a mistake in the presentation of the required reports, the nodes of the network can vote to determine its replacement (Zheng et al., 2017a).

3.4. Proof of Elapsed Time

Proof of Elapsed Time is introduced by Intel as one of the consensus methods of the blockchains that similar to the PoW, each miner is required to solve a hash problem. Each block approver (Miners) is selected in the shortest expected time and with respect to a reliable function due to block production. This election selects the miner randomly across the network and uses the Trusted Execution Environment (TEE) to ensure the safety of its electoral process. TEE is presented by specific Intel hardware (SGX, Secure Guard Extension). The main problem of this method is its dependence on Intel which conflicts with the blockchain philosophy base on decentralization (Salimitari & Chatterjee, 2018). In fact, we can classify this method as a semi-centralized consensus algorithm.

3.5. Practical Byzantine Fault Tolerance

This consensus method is used to solve the Byzantine General problem. Today's malicious attacks upon software have been increasingly common. The growing dependence of the industry and governments on online information services will make malicious attacks more attractive and make the consequences more serious. In addition, the number of software errors has been increased due to the growing size and the complexity of the software. Since malicious attacks and software errors can be a result of arbitrary (Byzantine) behavior of faulty nodes, the importance of the Practical Byzantine Fault Tolerance algorithm can be understood (Castro & Liskov, 1999). Therefore, this algorithm is a form of responsive machine mode. This service will be modeled as a mode machine, that is responsible for nodes in a decentralized system (Schneider, 1990). In this method, all nodes must participate in the voting process in order to add the next block, and the consensus is reached when more than two-thirds of the nodes have a favorable opinion of the block. The PBFT can withstand the behavior of one-third of the platforms normally. For example, in a system with a malicious node, at least four nodes must have an agreement to reach the correct end. Otherwise, agreement and consensus will not be reached. In this way, the consensus is achieved faster and is more economical compared to Proof of Work. Also, unlike the Proof of Stake, this method does not require any asset in the stake for the consensus process (Salimitari & Chatterjee, 2018).

In conclusion, energy efficiency and high throughput are considered as its advantages and some points such as few or no parameters available for being scalable and possible delays as the network should wait for all nodes' votes are noted as its disadvantages.

3.6. Delegated Byzantine Fault Tolerance

This method follows the rules of the PBFT, but it does not require the participation of all nodes in the process of voting to add a new block. A number of nodes are selected as representatives of other nodes and, based on a series of rules, follow a consensus process like the PBFT method (Salimitari & Chatterjee, 2018). In this method, some professional nodes are voted to record transactions for all nodes. This method is used in the NEO algorithm (Zheng et al., 2017a). It worth to mention that Delegated Byzantine Fault Tolerance is less probable to face delays than PBFT but limiting the number of the voters can threaten the decentralization of the network

3.7. Proof of Weight (PoWeight)

The Proof of Weight combines a wide range of slightly different consensus algorithms based on the Algorand consensus model (Buntinx, 2018). Algorand achieves agreement through a Byzantine agreement protocol that is able to scale users according to different parameters called weights (Gilad, Hemo, Micali, Vlachos, & Zeldovich, 2017). In a blockchain-based on Proof of Weight, a weight is attached to each user. The weight is calculated by different factors which would lead to different consensus algorithms around proof of weight. These factors are usually based on how much money a user has in his account. The network would remain secure as long as two-thirds or greater fractions of users are honest. The double-spend attacks also cannot threaten the safety of a Proof of Weight based network.

Algorand has similarities to the Proof of Stake algorithm. In PoS, the percent of tokens a user owns in the network determines the amount of the reward which would increase the profitability of discovering the next block. In PoWeight, a different weighted value would be used. Filecoin and Chia are examples of cryptocurrencies currently using PoWeight. Filecoin calculates the weighted factor by considering the amount of IPFS data that a user owns and calls this algorithm "Proof of Spacetime" (Protocol-Labs, 2017). Chia depends on Proof of Space and proof of time to achieve consensus. Proof of Reputation is also another weighted factor used in PoWeight Systems.

In summarize, the PoWeight algorithm brings considerable great customization and scalability, confirms transactions very quickly and is efficient in using a power supply. On the other hand, as participants do not receive rewards in this network, it is difficult to keep users incentivized to participate. Although generating passive revenue streams is not designed in PoWeight core, this issue can be addressed by developing creative solutions.

3.8. Proof of Burn (PoB)

Proof of burn is an alternative method for reaching an agreement in a blockchain network. The idea behind it is that miners should not waste energy or time to prove that they have done something difficult to do. In this algorithm, miners have to burn some of their already owned cryptocurrencies to get rewards. Burning here means that a user is required to send some cryptocurrency to an "eater address" to receive coins, tokens or mining privileges on the network (Bitcoin-Wiki, 2018). The money sent to an eater address is unrecoverable and no one can spend it again, so it is called burnt and is out of circulation. Just like the processes in PoW, burning coins is an expensive activity for the user but consumes no resources and energy. The only resource being used in PoB is the user's willingness to take a short-term loss to receive a long-term reward.

As mentioned, in the case of eater addresses, the address is generated randomly and is not associated with any private key. Not having any relation with any private key means that the money stored in an eater address is basically inaccessible and nobody can spend it. It should be noted that all of PoB cryptocurrencies require burning proof of work mined cryptocurrencies like bitcoin. Slimcoin (SLM) is a cryptocurrency that burns bitcoin as a mining method and consensus algorithm (P4Titan, 2014). The more coins a user burn the more chances she/he will get to find the next block. This is also similar to PoS in which the rich would most probably get richer.

To summarize its attributes, it is creating more stability as we know someone who risks a short-term loss and spends his money in this way, would stay in the network for a longer time to gain profits. Moreover, as there is no factor making the investors centralized, PoB enhances decentralization and creates a distributed network. On the other hand, burning PoW mined coins wastes energy and time. If one day the value of PoB coins becomes greater than the PoW burned coins, we could say that PoB is more energy-efficient than PoW and the wasted coins, energy and time would be somehow recovered.

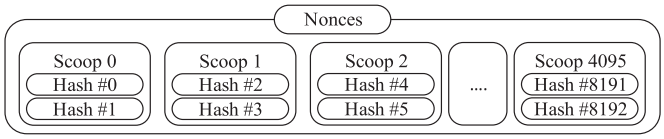

Fig. 2. The structure of blockchain-based on proof of capacity algorithm (Burstcoin, 2017).

3.9. Proof of Capacity

The concept of Proof of Capacity (PoC), also known as Proof of Space (PoSpace) was introduced by Dziembowski in 2015 (Dziembowski, Faust, Kolmogorov, & Pietrzak, 2015). Here, miners use the free spaces on their hard disk to mine free coins. The first cryptocurrency that utilized this algorithm was Burstcoin founded in 2014. The PoC algorithm consists of plotting the hard drive which means computing and storing solutions on your hard disk before the mining begins. Some solutions are faster than others. If a hard drive happens to have stored the fastest (closest) solution to the recent block's puzzle, then it wins the block.

In Burstcoin, implementing the PoC algorithm consists of two stages. The first stage is named plotting in which miners create something named "Nonce". Nonces are created by repeated hashing of data including miner's ID using a very slow hash function known as Shabal. As the Shabal hashes are hard to calculate, they are calculated in advance and are stored in the hard drive in the form of nonces. The more free space a miner allocates to plotting, the more nonces would be created. Nonces contain 8192 hashes. Every two hashes make a scoop so a nonce contains 4096 scoops labeled from 0 to 4095 as presented in Fig. 2. Before starting to mine, a miner should fill all of his/her desired free hard disk space with nonces. These nonces act like a lottery ticket that contains a series of numbers and letters. If one of the hashes in a nonce is the closest one to the recent puzzle in the network, it means to win the mining battle.

It should be noted that unlike bitcoin that needs special hardware such as ASICs and CPUs/GPUs for mining, the only hardware utilized in PoC is any regular Hard Disk Drive and therefore, no one can take advantage of special hardware. Moreover, using Hard Drives is said to be 30 times more energy-efficient than the ASIC based mining and there is no need to continuously upgrade your hardware, as an old Disk Drive can also store nonces. Additionally, as everyone has easy access to Hard Disk drives, the network would remain decentralized.

3.10. Proof of Importance

Proof of Importance (PoI) is another consensus algorithm that was first introduced in the NEM project in order to address criticisms in Proof of Stake algorithm. In PoS, the more node vests or holds an amount of currency, the more it will get scores for creating blocks as a reward. This method would incentivize the account holders to save coins instead of spreading them besides helping the rich get richer. In NEM's blockchain, however, each account or node is assigned an importance score which influences the account's chance of getting a small financial reward in exchange for adding users' transactions to the network. In order to be eligible to "harvesting" which means adding a block to the blockchain, an account should have at least 10,000 vested XEM (NEM, 2018).

Three factors which determine an account's overall score are mentioned as follows:

- Vesting: the higher the number of vested coins, the higher the score. Only coins count that have been in an account for a set number of days.
- Transaction partnership: who makes more transactions with other NEM accounts on the network would get a better score.
- Number and size of transactions in the last 30 days: each transaction above a minimum size would increase the account's score. Larger and more frequent transactions have a greater impact.

After calculating the account's score, the account will receive a chance related to the achieved score for adding a block to the blockchain network. This method ensures the decentralization of the blockchain while also makes a balance between locking up money in accounts and spreading it. In addition, according to the NEM's white paper, PoI is resistance to arbitrary manipulation by using NCDawareRank (Nikolakopoulos & Garofalakis, 2013), vested balance, decayed and weighted outlinks, and summation to unity in the calculation of importance score. These countermeasures make Proof of Importance resistant to Sybil-style attacks which are faulty or malicious presenting themselves as multiple identities in order to gain the control of a system. A loop attack, which is the act of sending XEM around via transfer transactions in a loop to boost importance score, is also confronted in PoI.

As a summary, the PoI algorithm can be fast and power-efficient, as no mining is needed and its scoring system will make it decentralized, scalable and safe. There is no special hardware needed to mine and is a considerable improvement to the traditional PoS algorithm.

3.11. Proof of Activity

Proof of Activity (PoA) is another common consensus algorithm and is introduced by Bentov et al. in 2014 (Bentov, Lee, Mizrahi, & Rosenfeld, 2014). The authors stated that they proposed a consensus algorithm based on the combination of PoW and PoS. It is an almost secure algorithm against possibly practical attacks on Bitcoin and has a low penalty regarding to network communication and storage space.

This algorithm is introduced as a guard against the potential problems in Bitcoin like the "tragedy of the commons" (Hardin, 1968), whereby miners begin to act only in their self-interests, and network attacks such as network denial-of-service and network isolation. For bitcoin, the tragedy of the commons can occur after all the 21 million mining-reward coins have been mined and the miners only receiving transactional rewards. As mentioned by Hardin (1968), "When the connectivity between the nodes is low, it becomes easier to deny service by flooding miner nodes or carry out a Sybil attack by isolating and transacting with some specific node" (Bentov et al., 2014).

Moreover, in Bitcoin an attacker may try to manipulate the price at the exchanges in which the Bitcoin is traded, to cause loss of confidence. However, with Proof of Stake based protocols, stakeholders are less likely to downward price spirals, because the coins that they hold generate revenue proportional to the actual commerce taking place. These potential problems in Bitcoin are mentioned in (Bentov et al., 2014), as a motivation to create the PoA algorithm by attempting to bring the best of both PoW and PoS.

In terms of security, the probability of 51% attack in the PoA algorithm drops to nearly zero, as such an attack would require the attacker to have 51% of all coins and also 51% of the mining power at the same time and therefore, PoA is more secure in comparison to PoW and PoS. On the other hand, PoA exploits the mining part of the PoW and therefore it needs a massive amount of energy and computational power. It is also mentioned that the PoA could be vulnerable to the double-spending attack (Bentov et al., 2014).

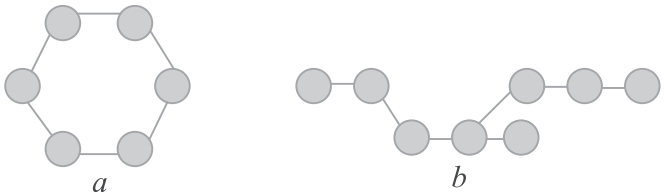

Fig. 3. (a) a cyclic and non-directed graph (b) an acyclic graph.

In conclusion, while securing the network against some potential problems like the 51% attack, the proof of activity has significant flaws. PoA requires a lot of resources and energy and also is susceptible to bribery based double-spending attacks (Bentov et al., 2014). Two popular cryptocurrencies, Decred, and Espers have adopted the PoA in their blockchain while Decred (DCR) has much better performance in terms of market pricing compare to Espers.

3.12. Directed Acyclic Graphs

Although Directed Acyclic Graphs or DAGs are basically a form of data structures and are not real blockchain networks, we decided to add them to this paper as they are widely being used in successful cryptocurrencies. Also knowing about the functions of DAG can help readers to better understand the blockchains. NXT, IOTA and IoT Chain are among the most successful applications of DAG. In real blockchain networks, transactions are stored in a chain of networks but in DAGs transactions are stored topologically in a graph. Fig. 3.a shows a cyclic and non-directed graph and Fig. 3.b presents an acyclic graph.

An acyclic graph is a graph that has no cycle. In an acyclic graph, information cannot be passed from one node to another node and return to the original node without encountering a node more than once. A Directed Acyclic Graph is an acyclic graph that the information can only pass through a pre-defined direction.

Due to their blockless structure, DAGs are considered as blockchains without blocks (Kotilevets et al., 2018). Cryptocurrency transactions are verified and added to the network in a way that is faster than PoW and PoS based networks, as there is no need to store them inside a real block and then verify the whole block (Benčić & Žarko, 2018). In a blockchain, an arbitrary period of time needs to be set to ensure the main chain remains viable. This waiting time is known as a block time and gives the network a time to consolidate and verify which branch of the chain is correct. However, in the DAGs, as long as the information is directed in the same way, nodes can exist in parallel.

This type of network would give us the ability to eliminate the need for block times and verify the transactions more quickly. The result is a fast, scalable and completely decentralized network (Kotilevets et al., 2018). Blockchains are susceptible to double spending, soft and even hard forks, however, in DAGs the validation of a certain transaction is decided by the number of the transactions behind it. This makes a DAG system faster and guarded against a double-spending attack.

In terms of the width of the network, adding the transaction to an earlier transaction every time would make the network too wide. On a DAG each validated transaction needs to link to an existing and new transaction of the network. When a transaction happens in a complete DAG network, the network would choose an existing later transaction to link to. This approach would keep the network width within a certain range that can support quick validation for transactions. IOTA has proposed its own algorithm

Table 1
Summary of different sets of performance evaluation criteria used in blockchain consensus literature.

Authors	Year	Performance evaluation criteria
(Croman et al., 2016)	2016	1- Maximum throughput, 2- Latency, 3-Bootstrap time, 4-Cost per Confirmed Transaction, 5-Transaction validation, 6-Bandwidth, 7-Storage
(Baliga, 2017)	2017	1-Transaction finality, 2-Transaction rate, 3-Token needed, 4-Cost of participation, 5-Scalability of the peer network, 6-Trust model, 7-Adversary Tolerance
(Mingxiao et al., 2017b)	2017	1-Byzantine fault tolerance, 2-Crash fault tolerance, 3-Verification speed, 4-Throughput (TPS), 5-Scalability
(Xu, Luthra, Cole, & Blakely, 2018)	2018	1-Architecture (Accounts, Transactions, and Contracts, State Management, Execution Environment), 2-Fault tolerance, 3- Economic Systems Analysis, 4-Block Size, 5-Block Time, 6-Transactional Throughput, 7-Block Throughput, 8-CPU Usage, 9-Transaction Size
(Nguyen & Kim, 2018)	2018	1-Energy efficiency, 2-Modern hardware, 3-Forking, 4-Double spending attack, 5-Block creating speed, 6-Pool mining
(Wang et al., 2018a)	2018	1-Origin of Hardness, 2-Implementation description, 3-ZKP Properties, 4-Simulation of random function, 5-Features of puzzle design, 6-Virtual mining, 7-Simulating Leader election
(Tang et al., 2019)	2019	1-Basic technology, 2-Applicability, 3-TPS, 4-Market capitalization, 5-Number of forks, 6-Total commits in GitHub, 7-Ranking in GitHub, 8-Team activity
(Alsunaidi & Alhaidari, 2019)	2019	1-Node Identity management, 2-Data Model, 3-Electing miners method, 4-Energy saving, 5-Tolerated power of the adversary, 6-Transaction fees, 7-Block reward, 8-Verification speed, 9-Throughput, 10-Block creation speed, 11-Scalability, 12-Extendible 13–51% Attack, 14-Double Spending, 15-Crash Fault Tolerance, 16-Byzantine fault tolerance
(Hasanova et al., 2019)	2019	1-Double spending attack, 2–51% attack, 3-Private key security,4-Noting at stake, 5-criminal problem, 6-selfish mining, 7-block withholding, 8-Bribery attack, 9-DDos/DoS, 10-Sybil attack, 11-Routing attack, 12-Time jacking attack
(Bano et al., 2019)	2019	1-Committee configuration, 2-Transaction censorship resistance, 3-DoS resistance, 4-Adversary model, 5-Throughput, 6-Scalable, 7-Latency, 8-Experimental setup

named Tangle to control the width of the network (Bramas, 2018; Sarfraz, Alam, Zeadally, & Khan, 2019).

There is no mining process on a DAG network and therefore, there is no dependency on special hardware hence power consumption is very low. The validation of the transactions happens almost instantly. The transaction fee could be very low and fast. So, this makes a DAG network friendly to small and even micro-transactions or payments. IoT chain, for example, can handle over 10,000 transactions per second. These characteristics of a DAG network along with its ability to defend against a 51% attack has made it a perfect approach for Internet-of-things and Machin-to-Machin communications (Bester, 2018).

4. Consensus algorithms evaluation criteria

With the extensive and intensive growth of blockchain technology and its application in different domains, a variety of complex consensus algorithms are developed which have unique, yet diverse properties and applications. The main purpose of this paper is to find the most important criteria which would affect the performance of these algorithms. With a comprehensive review of the literature, we identified a diverse set of criteria that have been applied to the different situations as presented in Table 1. In order to determine the most important criteria among the criteria sets, the paired comparison matrix method is applied. The pairwise comparison method is one of the most common methods to determine the importance or weight of each criterion. In this method, the criteria are compared with each other. This comparison is done base on determining the value of each criterionś preference over the other ones as shown in Table 2. The value of this preference is formed by the hierarchical decision-making method introduced by Saaty (2008) and the pairwise comparison method (Odu, 2019; Pamučar, Stević, & Sremac, 2018).

In order to determine the importance of the identified criteria, a questionnaire is designed and given to eight experts in this field. The results of these eight questionnaires are combined by geometric mean to form a final pairwise comparison matrix as presented in Table 3. The weight of identified criteria is considered as the geometric mean of each of the rows in the normalized pairwise comparison matrix. Finally, the consistency of the pairwise comparison matrix was determined by calculating the incompatibility rate that was less than 0.1. The obtained results are presented in Table 4 and the criterion with the value upper than 0.04 is considered as the most important metric for evaluating the performance of blockchain consensus algorithms.

In this section, we will do further analysis of these criteria and develop a framework to evaluate the consensus algorithms' performance based on the algorithms throughput, profitability of mining, degree of decentralization and consensus algorithms vulnerabilities as shown in Fig. 4. In the following, we would introduce the main criteria and their sub-categories that are introduced as a framework for consensus algorithms performance evaluation.

Table 2
The pairwise comparison scales among two criteria.

Types of preferences	Assigned value
Equal Importance	1
Weak Importance	2
Moderate importance	3
Moderate plus	4
Strong importance	5
Strong plus	6
Very strong	7
Very, very strong	8
Extreme importance	9

Table 3
Pairwise comparisons matrix.

	A1	*A2*	*A3*	…	*A20*
A1	1				
A2		1			
A3			1		
…				1	
A20					1

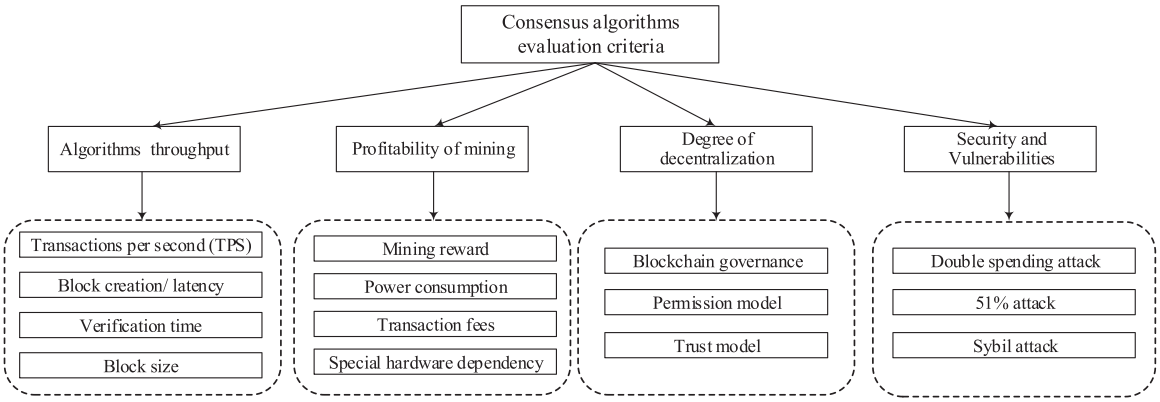

Fig. 4. A performance evaluation framework for blockchain consensus algorithms.

Table 4
The importance of identified criteria.

Performance evaluation criteria	Weight%	Performance evaluation criteria	Weight%
Transactions per second	8.68	Hardware dependency	4.41
Transaction fees	4.57	Time jacking attack	2.54
51% attack	8.03	Double spending attack	8.85
Latency	5.29	Number of forks	2.29
Governance	4.24	Verification time	5.37
Virtual mining	2.29	Permission model	5.48
Block size	4.18	Power consumption	7.52
Routing attack	2.38	Block withholding	2.87
Trust model	5.75	Mining reward	4.32
Crash Fault Tolerance	2.93	Sybil attack	7.99

4.1. Throughput

In the current financial system, customers have to wait a long time for payment validation. For example, an international transaction that is done by the current banking system can take up to 3 to 5 days or more. Blockchain technology as a decentralized and distributed system, can log and confirm any payments made without the need for an intermediary bank or person. This means faster transaction processing and lower transaction fees are possible with a blockchain system. The verification of the transactions is done with consensus algorithms. Consensus algorithms' throughput means the maximum rate of agreement on the values done in order to verify the transactions in a blockchain network (Bano et al., 2017). It means, the maximum throughput is the maximum rate at which the blockchain can confirm transactions (Croman et al., 2016), that is a trade-off between the maximum block size and the inter-block time. In the following section, we review several factors that influence consensus algorithm throughput.

4.1.1. Transaction per second (TPS)

Transaction per second that is commonly used for cryptocurrencies, is defined as the number of transactions executed per second. In other words, TPS is the number of transactions that occur in one second through an information system (Shi, Peng, Kou, & Chen, 2008). The transaction per second measurement is used to calculate the performance of systems that handle routine transactions and record-keeping jobs. Transaction per second is used to determine the speed of the platform or network in executing transactions. The higher the number of transactions per second, the faster the transactions will be executed, validated and confirmed on the same platform (Allwein, Schapire, & Singer, 2001). For instance, if a cryptocurrency executs12 transactions every minute, then the TPS for that is 0.2.

It tells us that if this cryptocurrency is able to complete 12 transactions in 60 s, the response time for each transaction will be 20 s. So, transaction per second is an important criterion in the blockchain network and depends on its consensus algorithm. Today's blockchains are designed for doing high TPS (Tang, Shi, & Dong, 2019). In order to provide an insight into the aforementioned consensus algorithms, Table 5 is presented to include some cryptocurrencies as a representative of each algorithm and compares them based on some criteria.

4.1.2. Block time/ latency

Block time or latency means the delay of verifying transactions and placing them in a block. The block is then entered into the blockchain and connected to the existing chain of blocks. In other words, latency means the time it takes from when a value is presented to the network, until when a consensus has been reached on it (Bano et al., 2017; Croman et al., 2016). As mentioned by Croman et al. (2016), there is also another factor that affects the validation process in blockchain networks named ***Bootstrap time***, which means "the time it takes a new node to download and process the history necessary to validate the current system state". Based on the available data (bitinfocharts, 2019), the average block times of some cryptocurrencies from Sep. 2018 till Aug. 2019 are presented in Table 6.

4.1.3. Block verification time

A transaction between the sender and receiver is said valid in a blockchain if it is requested by the sender. When a user makes a transaction, he (she) has to use his (her) private key as a digital signature (G.-T. Nguyen & Kim, 2018). When a transaction becomes valid, it will be included in a block and this block would be added to the blockchain network (Bano et al., 2017; Vallois & Guenane, 2017). Once a transaction is being included in a mined block, the transaction will receive one confirmation. With more subsequent mined blocks, the number of confirmations would increase for the transaction. Indeed, to reduce the risk of a double-spending attack, a number of block confirmations are needed before a transaction could be verified. In Table 7, we compare some consensus algorithms in terms of the verification speed based on their representative cryptocurrencies (Kraken, 2019; Mingxiao, Xiaofeng, Zhe, Xiangwei, & Qijun, 2017b).

4.1.4. Block size

A maximum transaction that a block can contain depends on the block size (Zheng, Xie, Dai, Chen, & Wang, 2017b). For example, Bitcoin's block size, for some safety reasons, is limited to 1 MB. The correlation between the block size and the number of transactions for the Bitcoin network is shown in Fig. 5.

Table 5
Comparison of consensus algorithms based on their representative cryptocurrencies (coincheckup, 2019).

Row	Consensus algorithms	Cryptocurrencies	Algorithm	Genesis Block	Rank	Market CAP ($)	TPS	Block Time Minutes	Mining reward
1	PoW	Bitcoin	SHA256	January 3, 2009	1	180,207,092,238	7	10	12.5 BTC
		Ethereum	Ethash (KECCAK256	July 30, 2015	2	22,757,000,420	15	0.25	2
		Litecoin	Scrypt	October 8, 2011	5	4,587,952,794	28	2.3	25
		Monero	Cryptonight	April 18, 2014	11	1,268,871,523	30	2	4.9
		Zcash	Equihash	October 28, 2016	28	348,443,197	27	2	10
2	PoS	Waves (LPoS)	LPoS	June 12, 2016	55	100,304,755	100	1	Non-minebable
		Qtum	POS 3.0	December 26, 2016	36	202,601,750	70	2	Non-minebable
		Nxt	SHA256	November 24, 2013	175	16,162,355	100	1	Non-minebable
		Blackcoin	Scrypt	February 24, 2014	500	4,569,548	0	1	Non-minebable
		Nano	Blake2b	February 29, 2016	45	123,741,646	7000	Instant	Non-minebable
3	DPoS	EOS	DPoS	July 1, 2017	7	3,641,735,649	4000	0.5	Non-minebable
		Cardano	Ouroboros (DPoS)	December 26, 2017	12	1,266,573,741	257	0.33	Non-minebable
		TRON	DPoS	August 28, 2017	13	1,186,299,015	2000	0.05	32 TRON
		Lisk	DPoS	January 30, 2016	47	118,714,644	3	0.284	Non-minebable
		BitShares	DPoS	July 19, 2014	58	91,575,735	100000	0.05	Non-minebable
4	PBFT	Ripple	N/A	April 11, 2013	3	12,010,477,031	1500	0.06	Non-minebable
		Stellar	N/A	April 6, 2016	10	1,410,189,643	1000	0.08	Non-minebable
		Zilliqa	Keccak	January 12, 2018	79	59,022,911	0	45s to 4 m	Non-minebable
5	PoC	Burst	Shabal256	August 11, 2014	190	14,417,212	80	4	460
6	DAG	IOTA	Curl-P	October 21, 2015	17	788,711,735	1000	Instant	Non-minebable
		Byteball (Obyte)	DAG	September 5, 2016	262	17,301,594	10	0.5	Non-minebable
		Travelflex	DAG	December 2, 2017	1374	163,648	3500	1	30.00 TRF
7	PoA (Hybrid PoW/PoS)	Dash	X11	January 19, 2014	16	850,165,302	56	2.5	2.09
		Decred	BLAKE256	December 15, 2015	32	233,089,579	14	5	18.22
		Komodo	Equihash	September 1, 2016	67	80,699,867	100	1	3.00 KMD
		Peercoin	SHA-256	August 19, 2012	373	7,844,163	0	10	37.36 PPC
		Espers	HMQ1725	April 28, 2016	1026	625,199	0	5	5000
8	dBFT	NEO	RIPEMD160	October 17, 2016	20	650,866,809	1000	0.25	Non-minebable
9	PoI	NEM (XEM)	Ed25519	March 31, 2015	26	403,570,701	10000	1	Non-minebable
10	PoB	Slimcoin	Dcrypt	May 07 2014	2661	16,195	0.00003	1.5	50.00 SLM

If there is no limitation for the block size, some miners can mine a large block while others cannot. This issue impairs the blockchain network. However, block size limits will have some issues:

- The slowdown in the network
- When the number of users and transactions increase, the speed of the network will be decreased.
- Higher transaction fees
- Because of the limitation of the block size, the verification of the transactions must be quick. This could be done by paying higher transaction fees. Hence, it would be a greater incentive for miners to process transactions with higher-paid fees (Asolo, 2018).

4.2. Profitability of mining

The profitability of mining is defined as the level of the net revenue that a validator or miner of the blockchain network gains in exchange for creating new blocks. It includes providing technical capacities for verifying the transactions and network tasks, resulting in a new data block on the network. There are different factors that affect the profitability of mining including the complexity of the process, mining rewards, power consumption, transaction fee and dependency of the mining process upon specific hardware.

4.2.1. Mining rewards

A consensus algorithm is defined as a process to achieve agreement on the value of data among multiple nodes in the distributed systems. In such systems, in order to obtain reliability and security in the network, the miners/validators are supposed to spend some sort of computational power, stake or hard disk space to verify the transactions and add new blocks to the blockchain and in return get some rewords. In fact, these rewards are considered as an incentive for miners to participate in the transaction validation process and consequently, the stability of the network is under the assumption that participants behave according to the system incentives (Kroll, Davey, & Felten, 2013). Based on the nature of designed consensus algorithms, the mining or validating rewards would vary in different blockchains. For example, based on the Bitcoin protocol the mining reward will be halved every 210,000 blocks. Most of the cryptocurrencies which are based on PoW algorithms are mineable, while cryptocurrencies based on PoS, DPoS, PBFT and DBFT like Ripple, Cardano, Stellar, NEM, and IOTA are pre-mined and there will be no mining reward in their network. It should be noted that the mining process is related to the difficulty of the network that is a measure of how difficult it is to find a hash below

Table 6
Average block times of some cryptocurrencies (Minutes) (bitinfocharts, 2019).

Cryptocurrency	Bitcoin	Ethereum	Litecoin	Ripple	EOS	Dash	Dogecoin	Blackcoin
Min	7.40	0.22	2.10	0.00	0.00	2.56	1.02	1.01
Max	14.50	0.34	3.09	0.00	0.00	2.69	1.06	1.19

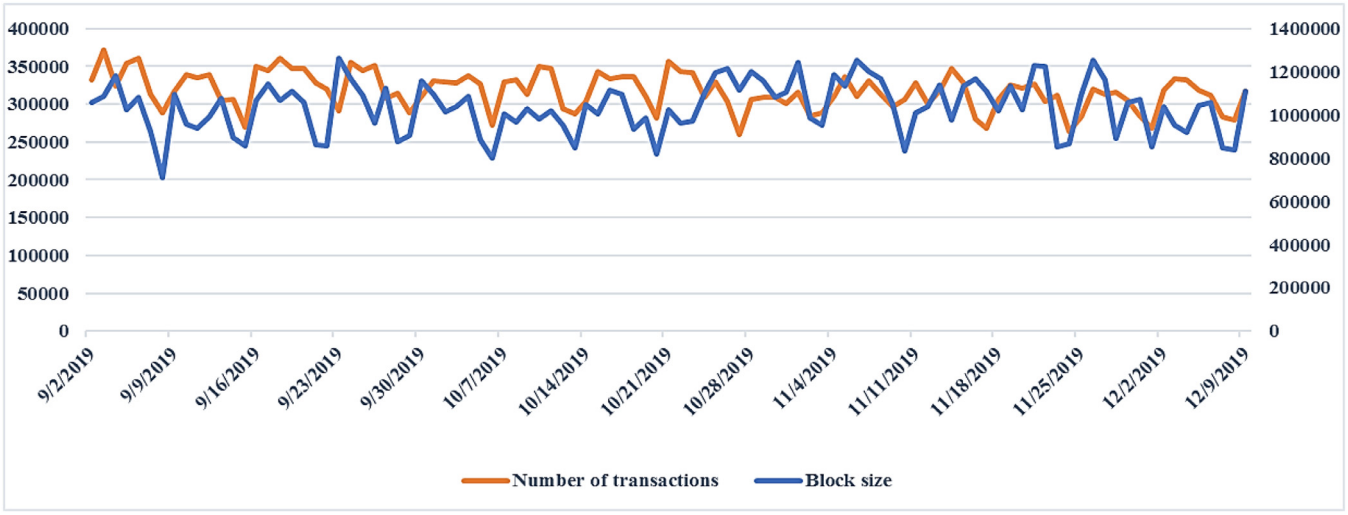

Fig. 5. The correlation between the block size and the number of transactions for the Bitcoin network (Blockchain, 2019a).

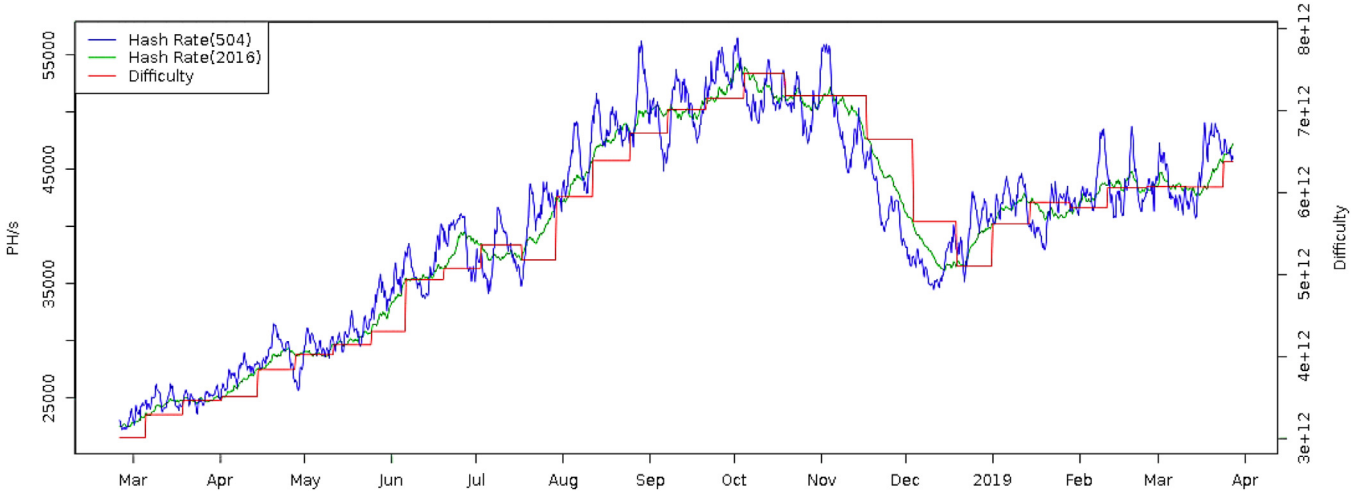

Fig. 6. The correlation between bitcoin hash rate and difficulty for the last 9 months (Bitcoinwisdom, 2019).

a given target. Hence, the difficulty of the network will be risen, by increasing the amount of hashing power that is applied to the mining process as shown in Fig. 6 for the bitcoin (Antonopoulos, 2014).

Table 7
Block confirmation time in different cryptocurrencies.

Cryptocurrency	Consensus algorithm	# of Confirmations	Validation Time (Minutes)
Augur (REP)	PoW	30	6
Bitcoin (XBT)		6	60
Bitcoin Cash (BCH)		15	150
Dogecoin (XDG)		20	20
Ether (ETH)		30	6
Litecoin (LTC)		12	30
Monero (XMR)		15	30
Zcash (ZEC)		24	60
Cosmos (ATOM)	PoS	N/A	Near-instant
Qtum (QTUM)		24	60
Dash (DASH)	PoA	6	15
Ripple (XRP)	PBFT	N/A	Near-instant
Stellar Lumens (XLM)		N/A	Near-instant
Cardano (ADA)	DPoS	15	10
Eos (EOS)		N/A	Near-instant

The competition for finding the next block and obtaining the reward is so high in such a way that an individual miner has a very little chance to be successful in a mining process, hence joining in some available mining pools to share the hashing power and the reward among the participants is an economical and reasonable strategy (Eyal & Sirer, 2018; W. Wang et al., 2018a). However, this kind of cooperation between the miners accumulates the power of mining in the hands of a few groups and decrease the decentralization level of the blockchain (Wang et al., 2018a). Fig. 7 shows the proportion of hash rate share (the mining power) among the most well-known bitcoin mining pools.

As we can see in Fig. 7, most of the amount of mining power is distributed among 13 mining pools. According to (Tuwiner, 2019), these mining pools are located in a few countries like China (81%) and Czech Republic (10%). Thus, this kind of power-sharing truly decreases the decentralization of the blockchain network.

4.2.2. Power consumption

One of the most important criteria that affect the assessment of blockchain consensus algorithms is their power consumption. To make it more clear, consider Bitcoin's PoW consensus algorithm. The authors in (Böhme, Christin, Edelman, & Moore, 2015; O'Dwyer & Malone, 2014) stated that the bitcoin's energy

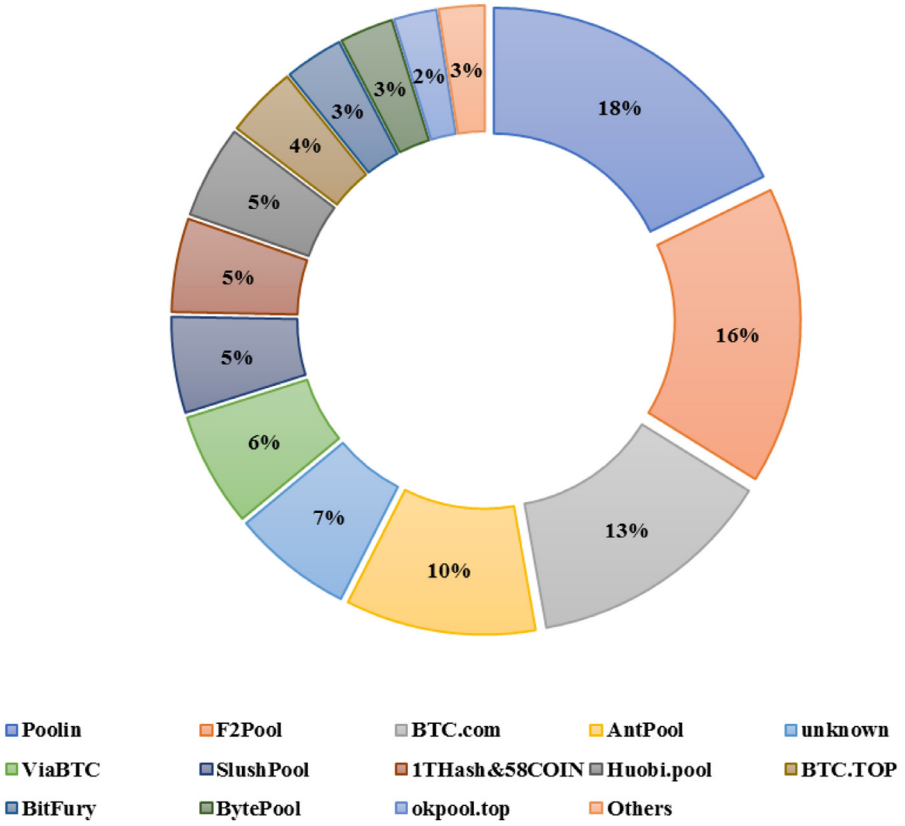

Fig 7. Hashrate distribution among the largest bitcoin mining pools (Blockchain, 2019b).

consumption can now power an entire country. This effect is not just about electricity consumption. The biggest problem is that Bitcoin's network is mostly fueled by coal-fired power plants in China (digiconomist, 2019). This results in extreme carbon footprints for every single Bitcoin transaction. A study published in Nature Climate Change suggests that this Bitcoins carbon emissions alone could push global warming above 2 °C in less than three decades (Mora et al., 2018), besides the authors in (Li, Li, Peng, Cui, & Wu, 2019) reported that the Monero network caused carbon emission of 19.12 to19.42 thousand tons in China from April to December 2018. It should be noted that hashing algorithms that are utilized in the consensus method mostly determine the mining efficiency. Here, the mining efficiency is defined as the ratio of the hashes per second, divided by the power consumed (Li et al., 2019). The hash function is mathematically considered as $y = H_F(x)$ where x is an input with different length and H_F is a hash function that generates a binary string of fixed length from any binary string of arbitrary length (Körber, Keller, & Holmbacka, 2018). It should be noted that cryptographic hash functions should be preimage resistant, second preimage resistant and collision resistant. The performance and energy consumption of some hashing functions such as SHA256, SHA512, RIPEMD160, BLAKE256, MD6, and Whirlpool are compared in (Damasevicius, Ziberkas, Stuikys, & Toldinas, 2012; Körber et al., 2018; Kumar Raghuvanshi, Khurana, & Bindal, 2014). Although SHA2 family including SHA256, which is used in Bitcoin, is proved to be unbreakable, they are not time efficient compared to SHA1 and MD5 (Kumar Raghuvanshi et al., 2014). On the other hand, Blake2 series of the cryptographic hash functions, which are being used in Nano and Siacoin, is faster and more secure than MD5, SHA-1, 2 and SHA-3 hash algorithms (Coinguides, 2018). A comparison of the most well-known hash function algorithms that are utilized in consensus algorithm, is presented in Table 8. To summarize, in blockchain domain we are obviously looking for more energy efficient consensus algorithms and this an important indicator for their evaluation.

4.2.3. Transaction fee

The transaction fee is a fee that is required to be paid to the miners to verify the block on the network which contains a particular transaction (Tang et al., 2019). Transaction speed and transaction fees are two closely-related concepts. Accordingly, high-value transactions are typically the quickest. If a user has paid a larger fee, cryptocurrencies miners will prioritize his/her payment over others. Fees typically increase in accordance with the growing usage and popularity of cryptocurrency networks (Zhang, Shi, Tian, & Zhu, 2009). In Fig. 8, the transaction fees of Bitcoin are shown.

As shown in Fig. 8 the transaction fee is an important part of Bitcoin as an incentive for its miners in facilitating the process of transactions But some other cryptocurrencies have shown that the transaction fee is not necessary (e.g. Zcahs, ripple, blackcoin, IOTA) and utilize other persuasive methods for the miners participation in the block creation process. In fact, these blockchains use some methods to eliminate transaction fees and provide a free user experience. These approaches are mentioned as follows:

- Charging developers

EOS blockchain does not consider the fee for transactions but instead, a new user should purchase RAM from block validators that are needed for making the transaction in the network.

- Using multiple tokens

NEO is a blockchain that uses two different tokens, NEO & GAS. The NEO token forms the investment aspect on the NEO platform. The GAS token works as permission for using NEO Blockchain.

- Making users do their own work

Table 8
A comparison of hash function algorithms utilized in consensus algorithms.

Hash function	Properties	Cryptocurrencies
SHA-256	SHA256 is a cryptographic hash function belonging to the Secure Hash Algorithm 2 family designed by the NSA. Its hash rates are at the GH/s range or higher and the block time with SHA-256 tends to be slower than Scrypt. Although it is more complex and uses higher energy for the mining process, it is resistant to security issues.	Bitcoin, Bitcoin Cash, Nxt, Peercoin, Namecoin
Scrypt	This is an encryption method that requires larger volumes of random access memory (RAM) compare to the SHA256, it is also quicker and has less energy consumption. The hash rates of scrypt based blockchains mostly range in the KH/s or MH/s areas of difficulty. It is designed to be ASIC resistant.	Litecoin, Dogecoin, MonaCoin, ReddCoin, BlackCoin
Ethash	Ethash is considered as the PoW algorithm in Ethereum-based blockchain. It is a modified version of Dagger-Hashimoto and the hash function used by Ethereum is Keccak-256 which is ASIC resistant and it is memory intensive algorithm.	Ethereum, Ethereum Classic, WhaleCoin
Blake2	BLAKE2 is a secure and fast cryptographic hash function based on Dan Bernstein's ChaCha stream cipher. It has two variants, Blake2b and Blake2s optimized for 64-bit and 32-bit platforms, respectively.	Siacoin, Nano, Decred, Verge
X11	X11 designed by Evan Duffield is a combination of eleven scientific hashing algorithms for the PoW that is specifically designed for GPU mining. This algorithm is a chain of 11 different scientific hashing functions including blake, bmw, groestl, jh, keccak, skein, luffa, cubehash, shavite, simd, echo which cause a high level of complexity, confidentiality, and safety for cryptocurrencies.	Dash, SmartCoin, Polis
Cryptonight	CryptoNight is a PoW hashing algorithm developed to be suitable for CPU and GPU minge and ASIC resistant. Not only it provides true anonymity, but also it pledges untraceable payments and unlinkable transactions.	Bytecoin, Monero, Electroneum, Dero

Some blockchains do not rely on specific users for doing the transaction verifications. Instead, every user needs to participate in the verification process. For example, in IOTA, a node must confirm two previous transactions from other users to get his own transaction verified.

- Subsidizing the platform

Kin is a cryptocurrency that was introduced by Kik. Kik has managed to eliminate transaction fees and was constructed in Ethereum blockchain. But the developer states that this blockchain is very slow and its fees are too high and for this reason, they decided to fork and transfer it in Steller blockchain. Kik can easily cover the cost of Kin transactions without charging transaction fees (Dalton, 2018).

4.2.4. Special hardware dependency

The process of mining the blocks is highly competitive and meanwhile, the difficulty rate of blockchain networks is considerably high, so that in most cases attempting to win this competition without specific hardware called "application-specific integrated circuit" abbreviated as ASIC, is impossible. In fact, the higher the hash rate is needed, the longer and more difficult the process would be that is defined as "network hash difficulty". As the hash difficulty grows exponentially, it needs more energy, time and resource dedication to engage in the mining process which is prohibitive for many individual miners. Hence, some consensus algorithms such as PoS, dPoS and Proof of Authority (PoA) are designed to be ASIC resistant. In some other algorithms, like proof of elapsed time, the consensus protocol is dependent official specific hardware such as Intel SGX and consequently, they prevent high resource utilization and high energy consumption. As all of the nodes are required to use this specific kind of hardware in this consensus algorithm, the process of mining would remain fair.

4.3. Decentralization levels

In the case of the decentralization of a blockchain, there are three major factors identified through reviewing the literature include the permission model, trust model, and blockchain governance model. In the following, these factors are discussed and some cryptocurrencies have been compared in terms of decentralization level (Chu & Wang, 2018).

4.3.1. Governance

Governance is one of the factors alongside the other criteria that impact the decentralization of a blockchain. Although it is not directly related to the consensus algorithm of a blockchain, it is a very important factor in defining the decentralization level of a cryptocurrency. For instance, governance in the Bitcoin means the process by which the rules of block verification are decided and set so that the individuals should adopt those rules for verifying the payments and blocks. Like that, in a traditional bank, these decisions are made by the bank holders.

As described in (De Filippi & Loveluck, 2016), Bitcoin's governance consists of two distinct coordination mechanisms: "governance by the infrastructures (achieved by the bitcoin protocol) and

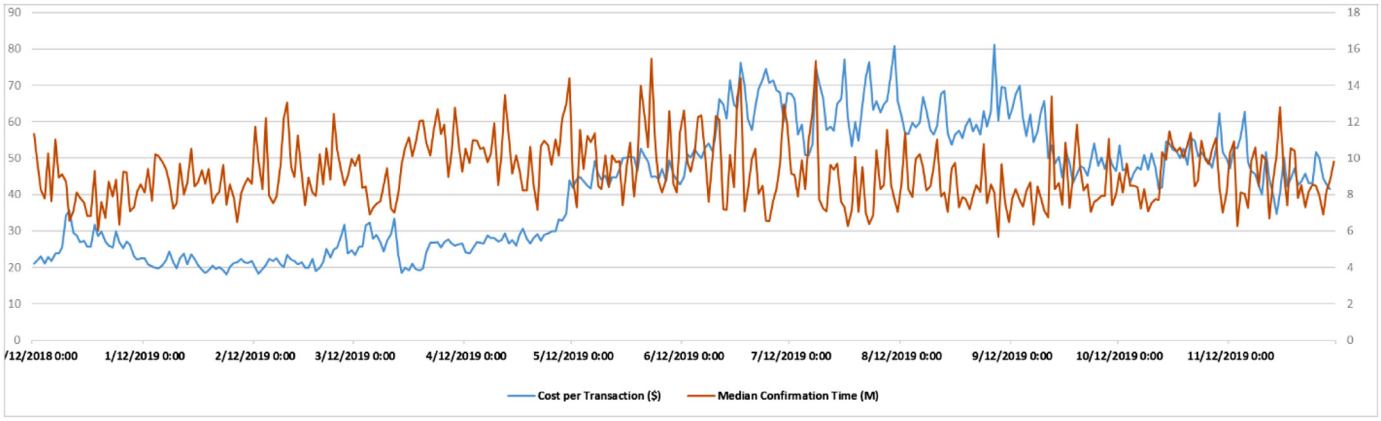

Fig. 8. Transaction fees of Bitcoin compared to the median confirmation time (Blockchain, 2019a).

Table 9
comparison of some cryptocurrencies in terms of permission models.

Permission Model	Trust Model	Cryptocurrency	Consensus Model
Permissioned	non-trustless	Neo	dBFT
	non-trustless	Icon	LFT (Loop Fault Tolerance)
	non-trustless	WTC	Hybrid PoW/PoS
	non-trustless	EOS	DPoS
	non-trustless	Ark	DPoS
	non-trustless	Lisk	DPoS
	non-trustless	VeChain	Proof-of-Authority
	non-trustless	Nuls	Proof-of-Credits
	trustless	Hashgraph	asynchronous BFT
Permissionless	trustless	Bitcoin	PoW
	trustless	Ethereum	PoW
	trustless	Litecoin	Pow
	trustless	Dogecoin	PoW
	trustless	Monero	PoW
	non-trustless	Stellar	PBFT
	non-trustless	XRP	N/A
	trustless	Nano	PoS
	trustless	Cardano	DPOS
	trustless	Decred	Hybrid PoW/PoS
	trustless	Zilliqa	PBFT
	trustless	Elastos	DPoS
	trustless	IOTA	DAG

governance of the infrastructures (managed by the community of developers and other stakeholders)". The authors stated that although bitcoin is an open-sourced project, the maintenance and development of the Bitcoin code thoroughly relies on a small group of skilled developers that play a key role in designing the platform. It means that the protocol of the Bitcoin is not completely automated and is not self-deciding its governance and has a technocratic power structure (De Filippi & Loveluck, 2016). This could potentially threaten the decentralization level of the blockchain. They also believe that a proper governance of Bitcoin structure that can fully grant a decentralized network can only be achieved by "publicly acknowledging its political dimensions, and replacing the current technocratic power structure of the Bitcoin project with an institutional framework capable of understanding (and accommodating) the politics inherent in each of its technical features" (De Filippi & Loveluck, 2016).

The previously discussed PoW based cryptocurrency was just an example of how the governance of the infrastructures in a consensus algorithm could make a technocratic power structure and threaten the decentralization of the network (Beck, Müller-Bloch, & King, 2018).

The governance in other cryptocurrencies with different consensus algorithms could also affect the decentralization of their blockchain. We couldn't find published scientific papers on the governance of other cryptocurrencies and we could say that is an important factor that has not received enough attention from cryptocurrencies researchers (Atzori, 2015).

4.3.2. Permission model

Except for public, private and federated blockchain as discussed in Section 1, there is another category that is needed to be taken into account as permissionless/open and permissioned/closed blockchain. Here, private and public blockchain refers to the degree of anonymity of validators. On the other hand, permissionless and permission blockchain refers to the degree of trust in validators. Hence, based on these two dimensions, four possible alternatives and different consensus algorithms are available for different business cases. In the following, the blockchain systems are discussed based on their permission model.

4.3.1.1. Permissioned system. In a permissioned system, the identity of the users is blacklisted through some types of a KYC (Know Your Customer) or KYB (Know your Business) procedure. KYC and KYB are common methods of managing identities in traditional finance (Swanson, 2015). In other words, in a permissioned platform, only specific agents are allowed to write on the blockchain ledger. It means in a permissioned platform the degree of trust in validator is higher than permissionless platforms.

4.3.1.2. Permissionless system. In contrast, we have a permissionless system in which anyone can write on the ledger of the blockchain. The identity of the participated node is either anonymous or pseudonymous. The number of the nodes in a Permissionless system is expected to be large and untrusted, as anyone could become a node in the network. The consensus algorithm in such a system is required to account for malicious activities like a Sybil attack (Baliga, 2017; Fischer et al., 1985). Proof of Work is an example of a public and permissionless algorithm. Everyone could become a node in a proof of work network but at first, he needs to prove that he has spent a significant amount of energy solving the PoW puzzle to verify his identity.

The effect of the permission model on the decentralization of the network is obvious. A permissioned system is likely making the blockchain more centralized, while a permissionless setup makes the network more decentralized by accepting more diverse nodes (Swanson, 2015). However, securing a permissionless blockchain and verifying the nodes and transactions could be challenging. Table 9 compares some cryptocurrencies in terms of permission models (Kramer, 2019).

4.3.3. Trust model

The trust model is defined as the level of access to read data in blockchains (Voshmgir & Kalinov, 2017). In the case of the trust model, blockchains are categorized into two models: Trustless and non-Trustless. Before the emergence of Bitcoin, every form of currency required a central authority, like a bank, that you had to trust in order to do a transaction with. Bitcoin was invented to change it. In bitcoin, if someone broadcasts a transaction, all of the nodes that receive it can validate the signatures and would discard the transaction if the signatures are not valid. Such a system, that a node does not need to trust anyone to do a transaction, is called a trustless system.

However, not every blockchain consider being fully trustless today. For example, at the date of writing this paper, Ripple (XRP)

locked 52% of the whole supplies of its network and only 6 accounts owned more than 51% of the circulation supply. So Ripple is holding the verification power for itself and to do a transaction, a node needs to trust these validators that are a part of the ripple website itself. However, this was just an example and these statistics might change in the future. According to the aforementioned criteria, the blockchain consensus algorism could be vary based on their applications as presented in Fig. 9.

As seen in Fig. 9, we can classify the permission and trust model of consensus algorithms in four types:

- Permissionless and public: Consensus algorithms like PoW are in this class. The best applications of this type are in Cryptocurrencies, Betting systems and Video Games. The decentralization of this system is its core competency.
- Permissioned and Public: PoS consensus algorithm is classified in this type. It is best for being used in Voting systems and Whistleblowers.
- Permissionless and Private: FBA algorithm sits in this class type. It will be used in SCM and Governmental financial records.
- Permissioned and Private: Algorithms like BPFT are under this section. Critical and sensitive confidential systems like Military, National defense and Law enforcement can utilize this class type.

4.4. Blockchain consensus vulnerabilities

Blockchain security depends on the robustness and strength of the consensus algorithm that is used to verify the transactions and the blocks (Ferrag et al., 2018). In this section, the most common cybersecurity attacks that theoretically can threaten nearly all types of consensus algorithms are discussed. There are also other types of attacks and vulnerabilities in blockchain protocols but these common and fundamental vulnerabilities should always receive the most attention in comparing different types of blockchains (Boireau, 2018).

4.4.1. Double spending attack

A double-spending attack occurs when a person tries to spend a specific amount of money on the blockchain twice (Zhang & Lee, 2019). This could happen when an attacker attempts to create a normal transaction to include in a block and then after some time, creates a fraudulent conflicting transaction and pushes it into a new forked fraudulent block, trying to revert the transaction he has made. The attacker should then try to extend the fraudulent branch of the network that he has created until the fraudulent branch is verified and accepted as the correct branch that includes the fraudulent transaction (Dasgupta, Shrein, & Gupta, 2019). Although different consensus algorithms try to mitigate this vulnerability and have a different mechanism to address it, double-spending cannot be completely avoided in blockchain systems and theoretically is possible to happen all the time (Hasanova, Baek, Shin, Cho, & Kim, 2019).

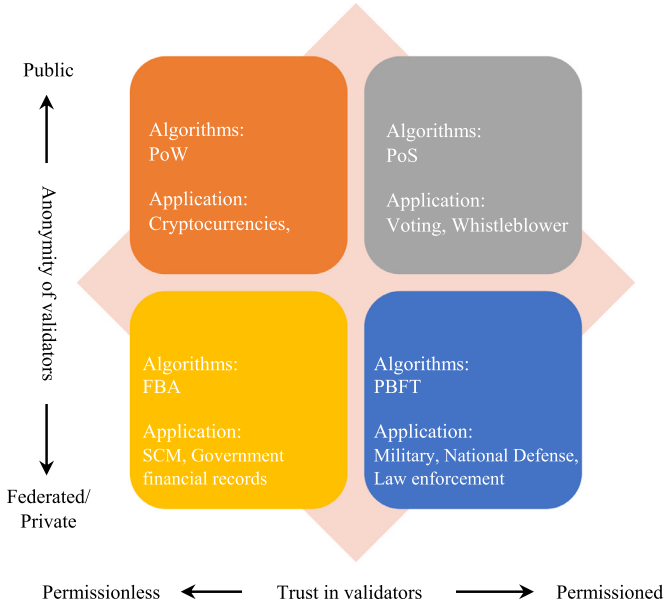

Fig. 9. Different types of blockchain-based on permission and trust models.

4.4.2. 51% attack

This type of attack was first exploited on bitcoin's PoW blockchain network, but can also be run on other blockchain systems. The 51% attack is also theoretically not avoidable (Bissias, Levine, Ozisik, & Andresen, 2016). The blockchain protocols try to raise the costs of this attack to defend it, but may not be able to completely prevent it. When an attacker is able to control more than 50% of the power (such as mining power or verification power) in the blockchain, he/she is able to do malicious activities like a double-spending or preventing other nodes from receiving their honest transactions. This type of attack is called a 51% attack (Conti, Kumar, Lal, & Ruj, 2018; Feng, He, Zeadally, Khan, & Kumar, 2018). The attacker does not always have to own 51% of the power of the network while he/she can bribe other nodes to follow him or he/she can temporarily rent the power he needs. In summary, this type of attack should always receive attention to comparing the blockchain's security. In terms of a comparison, (Sayeed & Marco-Gisbert, 2019) states that while acting differently against the 51% attack, PoW, PoS and DPoS algorithms are susceptible to it. However, PoA algorithm makes the cost of 51% attack higher since an attacker needs to have 51% of all coins and also 51% of the mining power at the same time.

4.4.3. Sybil attack

Sybil attack is a general form of an attack in which the attacker attempts to control a peer network by creating a number of fraudulent identities in the blockchain (Douceur, 2002). These identities appear to be unique users or nodes which are in fact in control of the attacker. These identities are used to gain voting power, block verification power, or even broadcasting a fake message in the social messaging network of the blockchain. A successful Sybil attack can grant the attacker a disproportionate control over the network or surround an honest node and try to influence the information reaching it and then gradually influence the ledger (Bissias et al., 2016).

Sybil attacks are hard to identify and prevent but blockchains try to deploy their own approaches to prevent it. Below are some of the approaches a blockchain may use for prevention (Mohaisen & Kim, 2013).

4.4.3.1. Increasing the cost of creating a node. Raising the cost of creating an identity is the first approach to mitigate the risk of a Sybil attack. For example, in the Proof-of-burn algorithm, users need to buy and then burn some coins – by sending some coins to a none-reversible address – to get verified as an identity (Bentov, Gabizon, & Mizrahi, 2016). In proof of stack, users need to have some coins in their stack and in the proof of work, users need to have and spend some computational power (Alachkar & Gaastra, 2018).

The challenge here is to find the ideal cost for the identity creation that efficiently mitigates the risk of a Sybil attack and also doesn't restrict normal people from joining the network. This cost might also change over time and a blockchain should always be ready to adopt the ideal cost. The cryptocurrency that is the

smartest to find this ideal cost and also is the fastest to change it depending on the needs of the network is the winner (Zhang & Lee, 2019).

4.4.3.2. Requiring some type of trust. The second common way to fight a Sybil attack is requiring a form of trust before letting a node joining the blockchain. This form of trust can become from a simple two-step verification email/SMS or asking for verification from a group of administrators. Doing some restrictions based on the IP address of the user and other common ways of defending against botnets are also used to provide this trust (Conti et al., 2018).

4.4.3.3. Giving unequal power to the identities. Giving different power to the users and nodes is another way to defend against a Sybil attack (Hasanova et al., 2019). For example, in the proof-of-weight algorithm, every account is given a weight based on several parameters such as the age of users' account, the number of unique users has made transactions with them, the amount of coins users owns and the number of transactions (Compare, 2019). This given weight gives every user a related amount of power to vote or participation. However, this unequal power distribution makes the system a meritocracy instead of a pure democracy and might not be interesting for new users (Sayeed & Marco-Gisbert, 2019).

Different cryptocurrencies use different approaches or different combinations of them to defend against a Sybil attack. There is no "Best" method and also there is no "Best" algorithm as each of them has its own pros and cons. However, when comparing the currencies, Sybil attack and effectiveness of the approaches to mitigate it should be evaluated carefully (Zhang & Lee, 2019).

5. Discussion and challenges

In recent years, the usage of blockchain technology is not limited to the digital cryptocurrency transmission, and the blockchain has found a variety of applications due to its high security, immutability and decentralized properties. As already stated, the core part of a blockchain is how to reach an agreement as a consensus algorithm. Different algorithms are developed for this intention and the most important ones are reviewed in Section 3. Plenty of studies have been done to investigate the efficiency, scalability, security, and applications of these algorithms in dealing with real-world problems.

In order to further evaluate the aforementioned consensus algorithms, we provided a summary comparison in Table 10 based on the information discussed in previous sections. As already mentioned, each of these algorithms is developed for a specific goal and application and has its own pros and cons. Some of them such as PoW, PoS, and DPoF are specifically designed to make it very impractical for an attacker to successfully carry out a Sybil attack. The goal of some others is to improve the shortcoming of well-known consensus algorithms, for example, PoI algorithms are designed to address the problems that can be found in the proof of stake model by identifying an account's overall support to the network.

As shown in Table 10, PoI consensus algorithm has very adequate performance among the other algorithms due to its safety against 51% and double-spending attacks, decentralized structure, energy-efficiency, and fast and scalable structure. Moreover, although DAG is not a consensus algorithm, the blockchains that are based on DAG have also very reasonable performance in every criterion. As a result, blockchains that are based on consensus algorithms should consider the DAG-based blockchains as a benchmark and compare their criteria with it. PoC has similar criteria to PoW and is improved in terms of speed and energy efficiency. Differently, dBFT has a more centralized structure compared to its similar algorithm, PBFT. Each other algorithm discussed in this table has its own pros and cons and none of them is still introduced as the best and final solution to the consensus problem.

In the rest of this section, an overview of current challenges in this domain is presented and the main unsolved problems of blockchain technology, particularly consensus algorithms, are discussed. For this purpose, we first analyze the available literature on consensus algorithms and consider the co-occurrence of the keywords of these studies.

The co-occurrence of the keywords can reflect the important points of the studies in the intended field and can be used as an outlook for future research (Bamakan, Nurgaliev, & Qu, 2019; Gu, Li, Li, & Liang, 2017). In the process of selecting the publications on blockchain's consensus algorithms, we received 148 papers and approximately 300 keywords. "VOS viewer" software was adopted for analyzing the co-occurrence of the keywords with each other. By receiving the extracted file of the papers' information, this tool illustrates the relationship mapping between the keywords. Fig. 10 illustrates the relationship mapping between the keywords of the extracted studies in the consensus algorithms field.

This figure is a graph representation of keywords and each circle as a top graph represents the keywords. The size of each circle shows the importance of keywords and the weight of each topic. The intention between two keywords represents the strength of the relationship between them. The shorter the intentions, the stronger the relationship would be. A curve between two words

Table 10
Summary comparison of blockchain consensus algorithms.

Consensus algorithms	Designing Goal	Decentralization level	Permission model/ Node Identity Management	Electing Miners/ verifiers Based on	Energy efficiency	Scalability	%51 Attack	Double Spendingattack	Hardware dependency	speed
PoW	Sybil-proof	Decentralized	Permissionless	Work (Hash)	No	Strong	Vulnerable	Vulnerable	Yes	Slow
PoS	Energy efficiency	Semi-centralized	Permissionless	Stake	Yes	Strong	Vulnerable	Difficult	No	Fast
DPoS	Organize PoS effectively	Semi-centralized	Both	Vote	Yes	Strong	Vulnerable	Vulnerable	No	Fast
PBFT	Remove software errors	Decentralized	Both	Vote	Yes	Low	Safe	Safe	No	Slow
PoC	Less energy than PoW	Decentralized	Permissionless	Work (Hash)	Fair	Strong	Vulnerable	Vulnerable	Yes	Slow
DAG	Speed and Scalability	Decentralized	Permissionless	N/A	Yes	Strong	Safe	Safe	No	Fast
PoA	Benefits of both Pos and PoW	Decentralized	Permissioned	Vote and work	No	Strong	Safe	Vulnerable	Yes	Fair
dBFT	Faster PBFT	Semi-centralized	Permissioned	Vote	Yes	Medium	Vulnerable	Vulnerable	No	Slow
PoI	Improve PoS	Decentralized	Permissionless	Importance scores	Yes	Strong	Safe	Safe	No	Fast
PoB	N/A	Decentralized	Permissionless	Burnt coins	No	Medium	Vulnerable	vulnerable	No	Fast

means that these two words have appeared with each other. The thicker curve means that the co-occurrence has appeared more frequently. Furthermore, some of the key phrases have been appeared pale to show the figure better.

In Fig. 10, similar color nodes have grouped in a cluster that mostly focuses on the same topics. Here the keywords have placed into six clusters. These clusters show the main trends of the research in the blockchain and consensus algorithms area (Li, An, Wang, Huang, & Gao, 2016). For better analyzing, these clusters with their keywords are shown in Table 11.

According to the main characteristics and orientation of clusters presented in Table 11, we named them as (1) *Blockchain technology adoption constraints and opportunities*; (2) *Blockchain-based cloud computing and AI applications*; (3) *Consensus problems in blockchain*; (4) *Blockchain smart grid interoperability*; (5) *Game theory analysis of blockchain solutions*; (6) *Blockchain security vulnerability*. The first cluster discusses adapting the blockchain as a tamper-proof system and considers the constraints in different domain such as healthcare (Dwivedi, Srivastava, Dhar, & Singh, 2019; McGhin, Choo, Liu, & He, 2019; Yue, Wang, Jin, Li, & Jiang, 2016), IoT (Khan & Salah, 2018; Makhdoom, Abolhasan, Abbas, & Ni, 2018), IIoT and Electronics manufacturing services (Bahga & Madisetti, 2016; Rathore, Kwon, & Park, 2019). The second cluster focuses on decentralized approaches to the fog and edge computing (Kumar, Saha, Rai, Thomas, & Kim, 2019; Li, Zhu, & Lin, 2019; Yang, Chen, & Xiang, 2018), scalability of cloud infrastructure and analyzing the state machine replication in blockchain-based cloud solutions (Nguyen, Pathirana, Ding, & Seneviratne, 2019; Tuli, Mahmud, Tuli, & Buyya, 2019; Xiong et al., 2019). Besides, the application of artificial intelligence to further improve the efficiency of resource allocation in blockchain systems is presented (Mamoshina et al., 2018; Marwala & Xing, 2018; Xu et al., 2017). The topic of the next cluster is reviewing the consensus algorithms as we already discussed in the present study. Energy transition, energy efficiency, and renewable energy are the main research trends in the fourth cluster that attract the attention of researchers to blockchain-based smart grid solutions (Aggarwal et al., 2019; Zhang, Wang, & Ding, 2019). Game theory, as an analytical tool can help decision-makers in uncertain and complex situations, has been utilized to investigate a vast range of problems in blockchain such as selfish mining, majority attack, reward allocation, and pool selection and a complete review on these issues are presented in (Liu et al., 2019). Research and studies grouped in the last cluster mainly focus on security and safety of blockchain technology includes the 51% and double-spending attacks, denial of service, Sybil attack and other open challenges as blockchains security vulnerability (Dasgupta et al., 2019; Hasanova et al., 2019; Wang, Shen, Li, Shao, & Yang, 2019).

As the results of Table 11 shows, the application of blockchain technology in artificial intelligence is an emerging topic that has been attracting proliferating studies recently (Krittanawong et al., 2020; Nassar, Salah, ur Rehman, Svetinovic, & Discovery, 2020). The integration of blockchain and artificial intelligence would yield to the rise of decentralized AI. In fact, by now all the AI process from model training to algorithm deployment and optimization are completely centralized. The centralized nature of AI, and

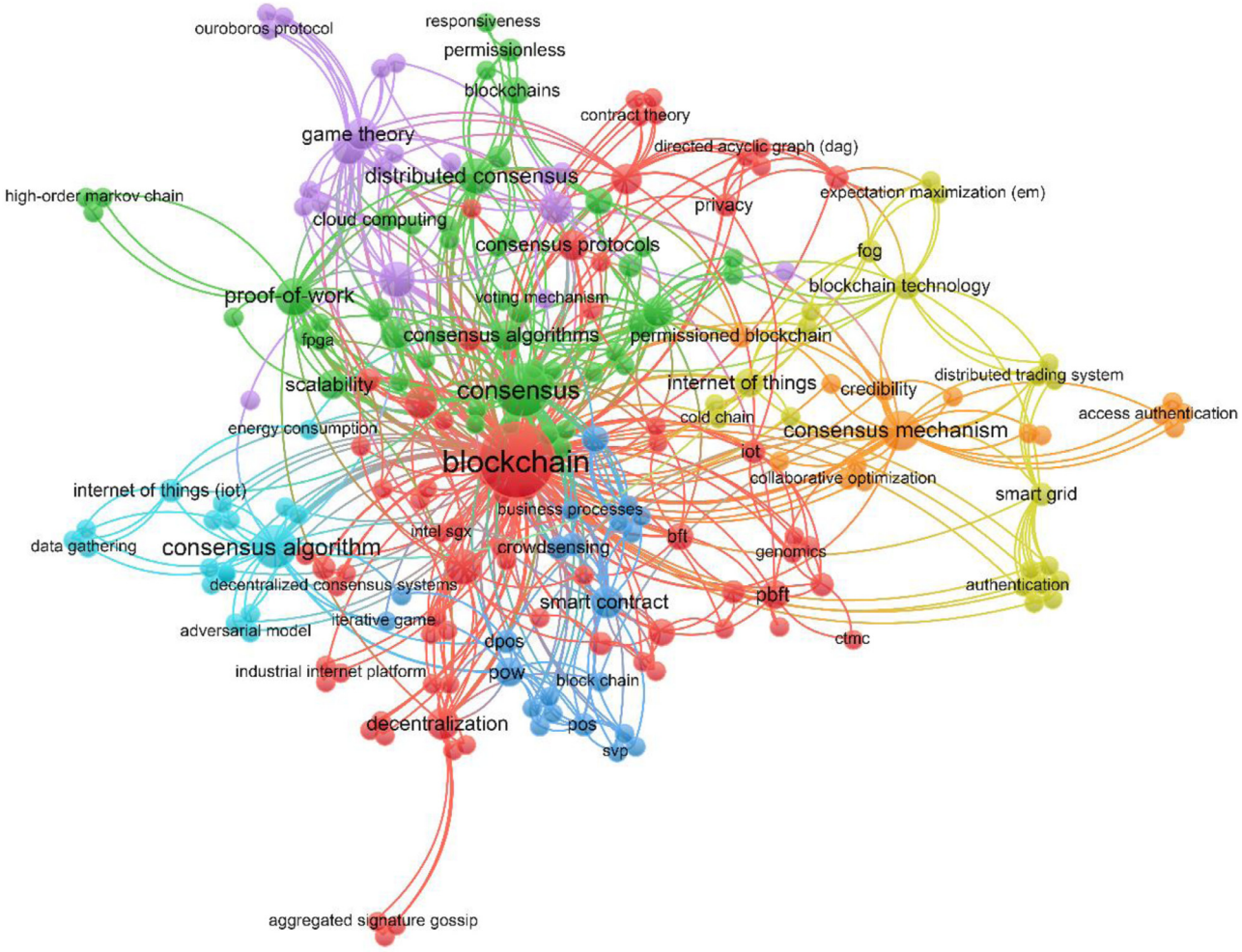

Fig. 10. The relationship mapping between the keywords of studies on consensus algorithms topic.

Table 11
Clusters of keywords and phrases used in consensus studies.

Clusters	Trends in blockchain studies	Keywords and phrases
1	Blockchain technology adoption constraints and opportunities	adoption constraints, aggregated signature gossip, anti-counterfeiting technology, channel state information, complexity, consensus algorithms, consistent hashing, contract theory, cross-tier interference, cryptographic sortition, data sharing, data tokenization, decentralization, device-to-device, digital currency, directed acyclic graph, energy consumption efficiency, genomics, healthcare, hedged learning, high throughput, industrial internet platform, industrial IoT (iIot), internet of vehicles, interoperability, low energy consumption, manufacturing service, collaboration, precision medicine, privacy, reputation management, stochastic reward nets, trusted execution environments
2	Blockchain-based cloud computing and AI applications	adaptive security, cloud computing, complex networks, consensus problem, distributed ledgers, elliptic curve cryptography, failure tolerance, fork, FPGA, group decision making, high-order Markov chain, hybrid consensus, iterative process, leaderless democracy, permissioned and permissionless, personalized difficulty adjustment, power system, resilience, responsiveness, round-robin, state machine replication, state reduction, trusted component, weighted voting, trusted oracles
3	Consensus problems in the blockchain	accountability, business processes, byzantine agreement, byzantine fault, crowdsensing, crowdsourcing, dbft, dpos, pos, pot, pow, quality control, rmbc, smart contract, smart manager, svp, UTXO, Paxos, hyper ledger fabric, intel sgx, proof of authority (poa), byzantine fault-tolerance, proof-of-qos (poq), ouroboros protocol
4	Blockchain smart grid interoperability	abstraction layer, Authentication, cold chain, computer architecture, distributed trading system, expectation-maximization, fog, gob, internet of things, java android, m2m, renewable energy, sign encryption, smart grid, smart home, supply chain, transactional energy
5	Game theory analysis of blockchain solutions	consensus propagation, consortium blockchain, dynamic leader election, enterprise services, game theory, incentive mechanisms, network delay, p2p networks, private blockchain, reward distribution, Shapley value, transaction evaluation, voting mechanism
6	Blockchain security vulnerability	block withholding attack, adversarial model, cyber-physical systems, cpss, data gathering, liveness, nash equilibrium, double-spending, pom: proof of majority, reducing energy, safety, security events detection, tendermint, Zero-determinant strategies, 51% attack, high attack tolerant

operating under the explicit trust boundaries of sole authority, would increase the possibility of data tampering and model or results manipulation. It means that there is no guarantee for the authenticity and integrity of collected data sources, which consequently raises the incompetency and the risk of centralized AI decision outcomes (Dinh & Thai, 2018; Nebula-AI, 2018). On the other hand, decentralized AI ensures that the process of data gathering, analyzing and the decision making happens on a decentralized, tamper-proof, cryptographically signed, and securely shared data blockchain-based framework which is developed on blockchain technology (Salah, Rehman, Nizamuddin, & Al-Fuqaha, 2019). Specifically, blockchain would bring more benefits to the new trends in AI such as explainable AI (Nassar et al., 2020; Rai, 2020), digital twins (Curry et al., 2020; Kaur, Mishra, & Maheshwari, 2020), automated machine learning (Podgorelec, Turkanović, & Karakatič, 2020), and lean and augmented data learning (Salah et al., 2019).

Artificial intelligence techniques and applications are generally designed to handle large amounts of data to aid humans for better decision making. Hence, the integration of AI and blockchain will facilitate the decentralized AI operations including decentralized storage, data management, learning model and model deployment (Salah et al., 2019). It is worth mentioning that centralized data storage, which mostly contains personal and sensitive data about users, are more vulnerable to attacks and security issues. Hence, a distributed data storage infrastructure based on blockchain technology will increase the security, facilitate the parallel data access from multiple nodes, and expand the scalability and reliability of the storage networks (Perard, Gicquel, & Lacan, 2019; Salah et al., 2019; Sultana et al., 2020). Furthermore, some researchers exclusively focus on a combination of machine learning, including deep and reinforcement learning, and blockchain technology. Blockchain can facilitate and secure data sharing when training and testing machine learning models. Moreover, the distributed nature of blockchain would help to save a tremendous amount of computational power while applying deep learning techniques. Also, blockchain can be utilized to optimize the parameter setting of machine learning techniques such as neural networks (Chen, Wan, Cai, & Cheng, 2019). Despite the attractiveness and importance of analyzing the role of blockchain technology in machine learning, few studies (Arora, Chopra, & Dixit, 2020; Chen et al., 2019; Dey, 2018; Kim, Kim, Hwang, & Seo, 2019; Wang, 2018; Winnicka & Kęsik, 2019) are done that further research in this topic would be of interest.

Despite recent advances in blockchain consensus algorithms, several other problems remain to be addressed as open challenges. In a blockchain, nodes should agree on a single shared state and this process is considerably slow and energy-consuming (Milutinovic, He, Wu, & Kanwal, 2016). Growing power consumption of blockchain networks is addressed by some researchers as a challenging limitation of this technology (De Vries, 2018; O'Dwyer & Malone, 2014). The authors in (Küfeoğlu & Özkuran, 2019) examined the energy consumption of bitcoin mining during December 2017 and reported its number between 1.3 and 14.8 GW which is between than installed capacities of Finland (~16 GW) and Denmark (~14 GW). In addition, (de Vries, 2019) discussed that even renewable energy is not able to address this issue and it seems that this energy consumption problem is still being heavily discussed both in business and academic communities and much more research is needed to find a promising solution.

As the cryptocurrencies and blockchain networks are becoming more popular and are being used for multiple applications, it has raised concerns about their ability to scale (Croman et al., 2016). As examined by (Poon & Dryja, 2016), Bitcoin as the most well-known cryptocurrency has a blockchain size of more than 150 GB and the maximum theoretical number of TPS of 7 which is too low compared to VISA, a traditional currency network, with the ability to process 20,000 transactions per second. Although some researchers addressed the scalability of blockchain technology, it is still an ongoing challenge in this field. The authors in (Zheng et al., 2017a), categorized the efforts proposed to address the scalability problem into two types: Storage optimization of blockchains and Redesigning the blockchain. Furthermore, the solutions are discussed as off-chain and on-chain solutions by (Bano, Al-Bassam, & Danezis, 2017). They argued that "Off-chain solutions allow for small and frequent transactions to take place over low-tier blockchain instances, parallel to and backed by the main blockchain" and On-chain solutions are the solutions that directly modify the blockchain design to increase its performance. In addition, (Dennis & Disso, 2019) claimed to conduct the first long term assessment of Bitcoin and Ethereum, model their growth over the next three years and propose a temporal blockchain

to increase the scalability. However, the scalability problem is now being heavily discussed among researchers and multiple approaches are being proposed to address it. Nevertheless, every solution has its own flaw-backs and no promising solution is still provided to fully address the scalability problem of the future blockchains.

It is worth mentioning that despite the rapid development of the technology in the realm of blockchain, there are limitations on the direct interactions and connections among the different blockchain networks. In fact, the problem is that this vast variety of blockchain networks works in an environment isolated from the others. This challenge is called "Interoperability" which is the ability to interact and recognize the information between different blockchain networks. In an interoperable environment, a user is able to transfer credits from one blockchain to another. There is a considerable amount of research done on the interoperability between databases (Sheth, 1999) but limited research exists on distributed ledger interoperability. For example, the authors in (Koens & Poll, 2019), presented an approach to examine the distributed ledgers' interoperability solutions and their sub-categories. Some commercial solutions such as Cosmos, Polkadot and Cardano are claimed to deal with this challenge, however, it is clear that we still need to do a massive amount of study in this area to find a practical solution to the interoperability problem of the current blockchains.

6. Conclusion

Over the past decade, we have been witnessing the exponential growth in the development of blockchain technology and its applications in different domains. In such distributed and decentralized systems, achieving agreement on a single data value among all participants to preserve the reliability of the system is a key challenging concern. It was the main objective of this paper to draw attention to the most well-known developed blockchain consensus algorithms. Hence, a brief overview of twelve consensus mechanisms and their advantages and disadvantages were discussed. Moreover, we proposed an analytic framework that consists of four different categories of criteria to evaluate the consensus algorithms' performance including algorithms throughput, the profitability of mining, degree of decentralization and algorithms' securities and vulnerabilities. Furthermore, we systematically analyzed the available literature on the consensus algorithms according to their main characteristics and orientation and then we provided the ongoing challenges and future research directions of this field.

Declaration of Competing Interest

We have no conflict of interest to declare.

References

Aggarwal, S., Chaudhary, R., Aujla, G. S., Kumar, N., Choo, K.-K. R., & Zomaya, A. Y. (2019). Blockchain for smart communities: Applications, challenges and opportunities. *Journal of Network and Computer Applications*.

Alachkar, K., & Gaastra, D. (2018). Blockchain-based Sybil Attack Mitigation: A Case Study of the I2P Network. https://www.os3.nl/_media/2017-2018/courses/rp2/p97_report.pdf.

Allwein, E. L., Schapire, R. E., & Singer, Y. (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research, 1*, 113–141.

Alsunaidi, S. J., & Alhaidari, F. A. (2019). Paper presented at the 2019 International Conference on Computer and Information Sciences (ICCIS). *A Survey of Consensus Algorithms for Blockchain Technology*.

Antonopoulos, A. M. (2014). *Mastering bitcoin: Unlocking digital cryptocurrencies*. O'Reilly Media, Inc.

Arora, M., Chopra, A. B., & Dixit, V. S. (2020). An approach to secure collaborative recommender system using artificial intelligence, deep learning, and blockchain *intelligent communication*. In *Control and Devices* (pp. 483–495). Springer.

Asolo, B. (2018). Block Size Explained. Retrieved August 4 2019, from https://www.mycryptopedia.com/block-size-explained/.

Atzori, M. (2015). Blockchain technology and decentralized governance: Is the state still necessary? *Available at SSRN, 2709713*.

Bach, L., Mihaljevic, B., & Zagar, M. (2018). Comparative analysis of blockchain consensus algorithms. *Paper presented at the 2018 41st International Convention on Information and Communication Technology*. Electronics and Microelectronics (MIPRO).

Bahga, A., & Madisetti, V. K. (2016). Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications, 9*(10), 533.

Baliga, A. (2017). Understanding blockchain consensus models *Persistent*. https://www.persistent.com/wp-content/uploads/2018/02/wp-understanding-blockchain-consensus-models.pdf.

Bamakan, S. M. H., Nurgaliev, I., & Qu, Q. (2019). Opinion leader detection: A methodological review. *Expert Systems with Applications, 115*, 200–222.

Bano, S., Al-Bassam, M., & Danezis, G. (2017). The road to scalable blockchain designs. *USENIX; login: Magazine*.

Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., & Danezis, G. (2017). Consensus in the age of blockchains. arXiv:1711.03936.

Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., et al. (2019). SoK: Consensus in the age of blockchains. In *Paper presented at the Proceedings of the 1st ACM Conference on Advances in Financial Technologies*.

Bashir, I. (2017). *Mastering blockchain*. Packt Publishing Ltd.

Beck, R., Müller-Bloch, C., & King, J. L. (2018). Governance in the blockchain economy: A framework and research agenda. *Journal of the Association for Information Systems, 19*(10), 1020–1034.

Benčić, F. M., & Žarko, I. P. (2018). Distributed ledger technology: Blockchain compared to directed acyclic graph. *Paper presented at the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*.

Bentov, I., Gabizon, A., & Mizrahi, A. (2016). Cryptocurrencies without proof of work. *Paper presented at the International Conference on Financial Cryptography and Data Security*.

Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of activity: Extending Bitcoin's Proof of Work via Proof of Stake. *IACR Cryptology ePrint Archive, 452 2014*.

Bester, N. (2018). Is Directed Acyclic Graph (DAG) Blockchain's New Competitor?*Retrieved March 15, 2019, from* https://www.investinblockchain.com/dag-blockchain-new-competitor/.

Bissias, G., Levine, B.N., .Ozisik, A.P., .& Andresen, G. (2016). An analysis of attacks on blockchain consensus. arXiv:1610.07985.

Bitcoin-Wiki. (2018). Proof of burn. *Retrieved March 30, 2019, from* https://en.bitcoin.it/wiki/Proof_of_burn.

Bitcoinwisdom. (2019). Bitcoin Difficulty. Retrieved 22 August 2019 from https://bitcoinwisdom.com/bitcoin/difficulty.

Bitinfocharts. (2019). Block time in some cryptocurrencies. *Retrieved* August 4 2019, from https://bitinfocharts.com/comparison/confirmationtime-btc-eth-xrp-xmr-zec-dash-doge-ppc-blk.html#1y.

Blockchain. (2019a). Blockchain Charts. *Retrieved* 20/12/2019 from https://www.blockchain.com/charts.

Blockchain. (2019b). An estimation of hashrate distribution amongst the largest mining pools. Retrieved from Retrieved 18/12/2019 from https://www.blockchain.com/en/pools.

Böhme, R., Christin, N., Edelman, B., & Moore, T. (2015). Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives, 29*(2), 213–238.

Boireau, O. (2018). Securing the blockchain against hackers. *Network Security, 2018*(1), 8–11.

Bramas, Q. (2018). The Stability and the Security of the Tangle. https://hal.archives-ouvertes.fr/hal-01716111/.

Buntinx, J. (2018). What Is Proof-of-Weight? *Retrieved March 31, 2019, from* https://nulltx.com/what-is-proof-of-weight/#.

Burrows, M. (2006). The Chubby lock service for loosely-coupled distributed systems. In *Paper presented at the Proceedings of the 7th symposium on Operating systems design and implementation*.

Burstcoin. (2017). Technical information to create plot files - Burst Coin. Retrieved 20/12/2019 from https://forums.getburst.net/t/technical-information-to-create-plot-files-burst-coin/914.

Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: current status, classification and open issues. *Telematics and Informatics, 36*, 55–81.

Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. *Paper presented at the OSDI*.

Chen, F., Wan, H., Cai, H., & Cheng, G. (2019). Machine Learning in/for Blockchain: Future and Challenges. arXiv:1909.06189.

Chu, S., & Wang, S. (2018). The Curses of Blockchain Decentralization. arXiv:1810.02937.

Coincheckup. (2019). Cryptocurrencies facts and figures. Retrieved 20/12/2019 from https://coincheckup.com/coins/bitcoin/analysis.

Coinguides. (2018). Blake2b Algorithm – List of Blake (2b) coins, miners and its hashrate. Retrieved from https://coinguides.org/blake2b/.

Compare, P. (2019). What Is Proof of Weight? Retrieved 19 August 2019, from https://coincodex.com/article/2617/what-is-proof-of-weight/.

Conti, M., Kumar, E. S., Lal, C., & Ruj, S. (2018). A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials, 20*(4), 3416–3452.

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., et al. (2016). On scaling decentralized blockchains. *Paper presented at the International Conference on Financial Cryptography and Data Security*.

Curry, E., Derguech, W., Hasan, S., Kouroupetroglou, C., ul Hassan, U., & Fabri-

tius, W. (2020). Building internet of things-enabled digital twins and intelligent applications using a real-time linked dataspace. In *Real-time linked dataspaces* (pp. 255–270). Springer.

Dalton, M. (2018). Feeless Cryptocurrency: How Do Blockchains Achieve Free Transactions? Retrieved 11/25/2019 from https://www.bitrates.com/news/p/feeless-cryptocurrency-how-do-blockchains-achieve-free-transactions.

Damasevicius, R., Ziberkas, G., Stuikys, V., & Toldinas, J. (2012). Energy consumption of hash functions. *Elektronika ir elektrotechnika, 18*(10), 81–84.

Dasgupta, D., Shrein, J. M., & Gupta, K. D. (2019). A survey of blockchain from security perspective. *Journal of Banking and Financial Technology, 3*(1), 1–17.

De Filippi, P., & Loveluck, B. (2016). The invisible politics of bitcoin: Governance crisis of a decentralized infrastructure. *Internet Policy Review, 5*(4).

De Vries, A. (2018). Bitcoin's growing energy problem. *Joule, 2*(5), 801–805.

De Vries, A. (2019). Renewable Energy Will Not Solve Bitcoin's Sustainability Problem. *Joule, 3*(4), 893–898.

Dennis, R., & Disso, J. P. (2019). An analysis into the scalability of Bitcoin and ethereum. *Paper presented at the Third International Congress on Information and Communication Technology*.

Dey, S. (2018). Securing majority-attack in blockchain using machine learning and algorithmic game theory: A Proof of Work. *Paper presented at the 2018 10th Computer Science and Electronic Engineering (CEEC)* 19-21 Sept. 2018.

Dib, O., Brousmiche, K.-. L., Durand, A., Thea, E., & Hamida, E. B. (2018). Consortium blockchains: Overview, applications and challenges. *International Journal On Advances in Telecommunications, 11* (1&2).

Digiconomist. (2019). Bitcoin Energy Consumption Index. Retrieved 1 August 2019 from https://digiconomist.net/BITCOIN-ENERGY-CONSUMPTION.

Dinh, T. N., & Thai, M. T. (2018). Ai and blockchain: A disruptive integration. *Computer, 51*(9), 48–53.

Doc, W. (2018). Leased Proof of Stake (LPoS). Retrieved from https://docs.wavesplatform.com/en/platform-features/leased-proof-of-stake-lpos.html.

Douceur, J. R. (2002). The sybil attack. *Paper presented at the International workshop on peer-to-peer systems*.

Drosatos, G., & Kaldoudi, E. (2019). Blockchain applications in the Biomedical Domain: A Scoping Review. *Computational and Structural Biotechnology Journal*. doi:10.1016/j.csbj.2019.01.010.

Dwivedi, A. D., Srivastava, G., Dhar, S., & Singh, R. (2019). A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors, 19*(2), 326.

Dwork, C., Lynch, N., & Stockmeyer, L. (1988). Consensus in the presence of partial synchrony. *Journal of the ACM (JACM), 35*(2), 288–323.

Dziembowski, S., Faust, S., Kolmogorov, V., & Pietrzak, K. (2015). Proofs of space. *Paper presented at the Annual Cryptology Conference*.

Eyal, I., & Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM, 61*(7), 95–102.

Feng, Q., He, D., Zeadally, S., Khan, M. K., & Kumar, N. (2018). A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications*.

Ferrag, M. A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., & Janicke, H. (2018). Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet of Things Journal, 6*(2), 2188–2204.

Fischer, M. J., Lynch, N. A., & Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM), 32*(2), 374–382.

Forum, W.E. (.2015). Deep Shift: Technology Tipping Points and Societal Impact. *Retrieved Feb. 10. 2019 From* https://www.weforum.org/reports/deep-shift-technology-tipping-points-and-societal-impact.

Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017). Algorand: Scaling byzantine agreements for cryptocurrencies. In *Paper presented at the Proceedings of the 26th Symposium on Operating Systems Principles*.

Gu, D., Li, J., Li, X., & Liang, C. (2017). Visualizing the knowledge structure and evolution of big data research in healthcare informatics. *International Journal of Medical Informatics, 98*, 22–32.

Gulzar, M. M., Rizvi, S. T. H., Javed, M. Y., Munir, U., & Asif, H. (2018). Multi-agent cooperative control consensus: A comparative review. *Electronics, 7*(2), 22.

Haber, S., & Stornetta, W. S. (1990). How to time-stamp a digital document. *Paper presented at the Conference on the Theory and Application of Cryptography*.

Hardin, G. (1968). The tragedy of the commons. *science, 162*(3859), 1243–1248.

Hasanova, H., Baek, U. j., Shin, M. g., Cho, K., & Kim, M. S. (2019). A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. *International Journal of Network Management, 29*(2), e2060.

Kaur, M. J., Mishra, V. P., & Maheshwari, P. (2020). *The convergence of digital twin, IoT, and machine learning: transforming data into action* digital twin technologies and smart cities (pp. 3–17). Springer.

Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems, 82*, 395–411.

Kim, H., Kim, S., Hwang, J. Y., & Seo, C. (2019). Efficient privacy-preserving machine learning for blockchain network. *IEEE Access, 7*, 136481–136495. doi:10.1109/ACCESS.2019.2940052.

Koens, T., & Poll, E. (2019). Assessing interoperability solutions for distributed ledgers. *Pervasive and Mobile Computing*, 101079.

Körber, O., Keller, J., & Holmbacka, S. (2018). Energy-efficient execution of cryptographic hash functions on big. LITTLE architecture. *Paper presented at the 2018 13th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*.

Korpela, K., Hallikas, J., & Dahlberg, T. (2017). Digital supply chain transformation toward blockchain integration. *Paper presented at the proceedings of the 50th Hawaii international conference on system sciences*.

Kotilevets, I., Ivanova, I., Romanov, I., Magomedov, S., Nikonov, V., & Pavelev, S. (2018). Implementation of directed acyclic graph in blockchain network to improve security and speed of transactions. *IFAC-PapersOnLine, 51*(30), 693–696.

Kraken. (2019). Cryptocurrency deposit processing times. Retrieved August 6, 2019, from https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times.

Kramer, M. (2019). What is the most decentralized cryptocurrency. Retrieved from Retrieved August 5 2019, from https://www.quora.com/What-is-the-most-decentralized-cryptocurrency.

Krittanawong, C., Rogers, A.J., Aydar, M., Choi, E., Johnson, K.W., Wang, Z. et al. (2020). Integrating blockchain technology with artificial intelligence for cardiovascular medicine. 17(1), 1–3.

Kroll, J. A., Davey, I. C., & Felten, E. W. (2013). The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In *Paper presented at the Proceedings of WEIS*.

Kshetri, N. (2018). 1 Blockchain's roles in meeting key supply chain management objectives. *International Journal of Information Management, 39*, 80–89. doi:10.1016/j.ijinfomgt.2017.12.005.

Küfeoğlu, S., & Özkuran, M. (2019). Energy Consumption of Bitcoin Mining. *Cambridge Working Papers in Economics 1948*. University of Cambridge: Faculty of Economics https://ideas.repec.org/p/cam/camdae/1948.html.

Kumar Raghuvanshi, K., Khurana, P., & Bindal, P. (2014). Study and comparative analysis of different hash algorithm. *Journal of Engineering Computers & Applied Sciences, 3*, 1–3.

Kumar, G., Saha, R., Rai, M. K., Thomas, R., & Kim, T. (2019). Proof-of-Work consensus approach in blockchain technology for cloud and fog computing using maximization-factorization statistics. *IEEE Internet of Things Journal, 6*(4), 6835–6842. doi:10.1109/JIOT.2019.2911969.

Lamport, L. (1998). The part-time parliament. *ACM Transactions on Computer Systems (TOCS), 16*(2), 133–169.

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS), 4*(3), 382–401.

Larimer, D. (2014). Delegated proof-of-stake (dpos). *Bitshare whitepaper, Retrieved March 31, 2019, from* http://docs.bitshares.org/bitshares/dpos.html.

Li, H., An, H., Wang, Y., Huang, J., & Gao, X. (2016). Evolutionary features of academic articles co-keyword network and keywords co-occurrence network: Based on two-mode affiliation network. *Physica A: Statistical Mechanics and its Application, 450*, 657–669.

Li, J., Li, N., Peng, J., Cui, H., & Wu, Z. (2019). Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies. *Energy, 168*, 160–168.

Li, M., Zhu, L., & Lin, X. (2019). Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing. *IEEE Internet of Things Journal, 6*(3), 4573–4584. doi:10.1109/JIOT.2018.2868076.

Liu, Z., Luong, N.C., .Wang, W., Niyato, D., Wang, P., Liang, Y.-.C. et al. (2019). A Survey on Applications of Game Theory in Blockchain. arXiv:1902.10865.

Macrinici, D., Cartofeanu, C., & Gao, S. (2018). Smart contract applications within blockchain technology: A systematic mapping study. *Telematics and Informatics*.

Makhdoom, I., Abolhasan, M., Abbas, H., & Ni, W. (2018). Blockchain's adoption in IoT: The challenges, and a way forward. *Journal of Network and Computer Applications*.

Mamoshina, P., Ojomoko, L., Yanovich, Y., Ostrovski, A., Botezatu, A., Prikhodko, P., … Zhavoronkov, A. (2018). Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare. *Oncotarget, 9*(5), 5665–5690. doi:10.18632/oncotarget.22345.

Marwala, T., & Xing, B. (2018). Blockchain and artificial intelligence. arXiv:1802.04451.

McGhin, T., Choo, K.-K. R., Liu, C. Z., & He, D. (2019). Blockchain in healthcare applications: Research challenges and opportunities. *Journal of Network and Computer Applications*.

Milutinovic, M., He, W., Wu, H., & Kanwal, M. (2016). Proof of luck: An efficient blockchain consensus protocol. *Paper presented at the proceedings of the 1st Workshop on System Software for Trusted Execution*.

Min, H. (2019). Blockchain technology for enhancing supply chain resilience. *Business Horizons, 62*(1), 35–45. doi:10.1016/j.bushor.2018.08.012.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017a). A review on consensus algorithm of blockchain. *Paper presented at the Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017b). A review on consensus algorithm of blockchain. *Paper presented at the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*.

Mohaisen, A., & Kim, J. (2013). The sybil attacks and defenses: A survey. arXiv:1312.6349.

Mora, C., Rollins, R. L., Taladay, K., Kantar, M. B., Chock, M. K., Shimada, M., et al. (2018). Bitcoin emissions alone could push global warming above 2 C. *Nature Climate Change, 8*(11), 931.

Nakamoto, S. (2019). Bitcoin: A peer-to-peer electronic cash system. *Manubot*.

Nassar, M., Salah, K., ur Rehman, M. H., & Svetinovic, D. (2020). Blockchain for explainable and trustworthy artificial intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10*(1), e1340.

Nebula, -A.I. (2018). NEBULA AI (NBAI)—Decentralized Ai Blockchain Whitepaper. https://www.chainwhy.com/upload/default/20180626/97a106c8ee2793553169caa9bee86cca.pdf.

NEM, T. (2018). Nem technical reference. URL https://nem.io/wpcontent/themes/nem/files/NEM_techRef.pdf.

Nguyen, D. C., Pathirana, P. N., Ding, M., & Seneviratne, A. (2019). Blockchain for secure EHRs sharing of mobile cloud based E-Health systems. *IEEE Access, 7*, 66792–66806. doi:10.1109/ACCESS.2019.2917555.

Nguyen, G.-. T., & Kim, K. (2018). A survey about consensus algorithms used in blockchain. *Journal of Information processing systems, 14*(1).

Nikolakopoulos, A. N., & Garofalakis, J. D. (2013). NCDawareRank: A novel ranking method that exploits the decomposable structure of the web. In *Paper presented at the Proceedings of the sixth ACM international conference on Web search and data mining*.

Odu, G. (2019). Weighting methods for multi-criteria decision making technique. *Journal of Applied Sciences and Environmental Management, 23*(8), 1449–1457.

O'Dwyer, K.J., & Malone, D. (2014). Bitcoin mining and its energy footprint. https://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0699.

Oki, B. M., & Liskov, B. H. (1988). Viewstamped replication: A new primary copy method to support highly-available distributed systems. In *Paper presented at the Proceedings of the seventh annual ACM Symposium on Principles of distributed computing*.

Ongaro, D., & Ousterhout, J. K. (2014). In search of an understandable consensus algorithm. *Paper presented at the USENIX Annual Technical Conference*.

P4Titan. (2014). A Peer-to-Peer Crypto-Currency with Proof-of-Burn. *Retrieved March 10, 2019, from* https://github.com/slimcoin-project/slimcoin-project.github.io/raw/master/whitepaperSLM.pdf.

Pamučar, D., Stević, Ž., & Sremac, S. (2018). A new model for determining weight coefficients of criteria in mcdm models: Full consistency method (fucom). *Symmetry, 10*(9), 393.

Panarello, A., Tapas, N., Merlino, G., Longo, F., & Puliafito, A. (2018). Blockchain and iot integration: A systematic survey. *Sensors (Basel), 18*(8). doi:10.3390/s18082575.

Pease, M., Shostak, R., & Lamport, L. (1980). Reaching agreement in the presence of faults. *Journal of the ACM (JACM), 27*(2), 228–234.

Perard, D., Gicquel, L., & Lacan, J. (2019). BlockHouse: Blockchain-based Distributed Storehouse System. In *2019 9th Latin-American Symposium on Dependable Computing (LADC)* (pp. 1–4). IEEE.

Podgorelec, B., Turkanović, M., & Karakatič, S.J.S. (2020). A Machine Learning-Based Method for Automated Blockchain Transaction Signing Including Personalized Anomaly Detection. 20(1), 147.

Poon, J., & Dryja, T. (2016). The bitcoin lightning network: Scalable off-chain instant payments. https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf.

Protocol-Labs. (2017). Filecoin: A Decentralized Storage Network. *Retrieved March 31, 2019, from* https://filecoin.io/filecoin.pdf.

Rai, A. (2020). Explainable AI: from black box to glass box. *Journal of the Academy of Marketing Science, 48*(1), 137–141.

Rathore, S., Kwon, B. W., & Park, J. H. (2019). BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network. *Journal of Network and Computer Applications, 143*, 167–177.

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International journal of services sciences, 1*(1), 83–98.

Salah, K., Rehman, M. H. U., Nizamuddin, N., & Al-Fuqaha, A. (2019). Blockchain for AI: Review and open research challenges. *IEEE Access, 7*, 10127–10149.

Salimitari, M., & Chatterjee, M. (2018). An Overview of Blockchain and Consensus Protocols for IoT Networks. arXiv:1809.05613.

Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *Paper presented at the Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on*.

Sarfraz, U., Alam, M., Zeadally, S., & Khan, A. (2019). Privacy aware IOTA ledger: Decentralized mixing and unlinkable IOTA transactions. *Computer Networks, 148*, 361–372.

Sayeed, S., & Marco-Gisbert, H. (2019). Assessing blockchain consensus and security mechanisms against the 51% Attack. *Applied Sciences, 9*(9), 1788.

Schenato, L., & Gamba, G. (2007). A distributed consensus protocol for clock synchronization in wireless sensor network. In *2007 46th IEEE Conference on Decision and Control* (pp. 2289–2294). IEEE.

Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys (CSUR), 22*(4), 299–319.

Sheth, A. P. (1999). Changing focus on interoperability in information systems: From system, syntax, structure to semantics. In *Interoperating geographic information systems* (pp. 5–29). Springer.

Shi, Y., Peng, Y., Kou, G., & Chen, Z. (2008). Introduction to data mining techniques via multiple criteria optimization approaches and applications. *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications, 3*.

Sultana, T., Almogren, A., Akbar, M., Zuair, M., Ullah, I., & Javaid, N. (2020). Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices. *Applied Sciences, 10*(2), 488.

Swanson, T. (2015). Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. *Report, available online*. https://allquantor.at/blockchainbib/pdf/swanson2015consensus.pdf.

Tang, H., Shi, Y., & Dong, P. (2019). Public blockchain evaluation using entropy and TOPSIS. *Expert Systems with Applications, 117*, 204–210.

Tuli, S., Mahmud, R., Tuli, S., & Buyya, R. (2019). Fogbus: A blockchain-based lightweight framework for edge and fog computing. *Journal of Systems and Software*.

Turner, B. (2007). The paxos family of consensus protocols. http://www.fractalscape.org/files/paxos-family.pdf.

Tuwiner, J. (2019). Bitcoin Mining Pools. Retrieved from 18/12/2019 from https://www.buybitcoinworldwide.com/mining/pools/.

Vallois, V., & Guenane, F. A. (2017). *Paper presented at the 2017 1st Cyber Security in Networking Conference (CSNet)*.

Voshmgir, S., & Kalinov, V. (2017). Blockchain, a beginners guide. *BlockchainHub, 30 September*.

Wang, L., Shen, X., Li, J., Shao, J., & Yang, Y. (2019). Cryptographic primitives in blockchains. *Journal of Network and Computer Applications, 127*, 43–58.

Wang, T. (2018). A Unified Analytical Framework for Trustable Machine Learning and Automation Running with Blockchain. *Paper presented at the 2018 IEEE International Conference on Big Data (Big Data)* 10-13 Dec. 2018.

Wang, W., Hoang, D.T., Xiong, Z., Niyato, D., Wang, P., Hu, P. Wen, Y. (2018b). A survey on consensus mechanisms and mining management in blockchain networks. *arXiv preprint* arXiv:1805.02707, 1–33.

Wang, W., Hoang, D.T., Xiong, Z., Niyato, D., Wang, P., Hu, P. et al. (2018a). A survey on consensus mechanisms and mining management in blockchain networks. arXiv:1805.02707, 1–33.

Winnicka, A., & Kęsik, K. (2019). Idea of using blockchain technique for choosing the best configuration of weights in neural networks. *Algorithms, 12*(8), 163.

Xiong, Z., Feng, S., Wang, W., Niyato, D., Wang, P., & Han, Z. (2019). Cloud/fog computing resource management and pricing for blockchain networks. *IEEE Internet of Things Journal, 6*(3), 4585–4600. doi:10.1109/JIOT.2018.2871706.

Xu, B., Luthra, D., Cole, Z., & Blakely, N. (2018). Eos: An architectural, performance, and economic analysis: Bitmex. Retrieved from https://www.whiteblock.io/library/eos-test-report.pdf.

Xu, C., Wang, K., & Guo, M. (2017). Intelligent resource management in blockchain-based cloud datacenters. *IEEE Cloud Computing, 4*(6), 50–59.

Yang, C., Chen, X., & Xiang, Y. (2018). Blockchain-based publicly verifiable data deletion scheme for cloud storage. *Journal of Network and Computer Applications, 103*, 185–193.

Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. *Journal of Medical Systems, 40*(10), 218. doi:10.1007/s10916-016-0574-6.

Zhang, D., Shi, Y., Tian, Y., & Zhu, M. (2009). A class of classification and regression methods by multiobjective programming. *Frontiers of Computer Science in China, 3*(2), 192–204.

Zhang, H., Wang, J., & Ding, Y. (2019). Blockchain-based decentralized and secure keyless signature scheme for smart grid. *Energy, 180*, 955–967.

Zhang, S., & Lee, J.-. H. (2019). Double-spending with a Sybil attack in the bitcoin decentralized network. *IEEE Transactions on Industrial Informatics*.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017a). An overview of blockchain technology: Architecture, consensus, and future trends. *Paper presented at the Big Data (BigData Congress), 2017 IEEE International Congress on*.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017b). An overview of blockchain technology: Architecture, consensus, and future trends. *Paper presented at the 2017 IEEE International Congress on Big Data (BigData Congress)*.