algorithms

MDPI

Article

Reputation-Driven Dynamic Node Consensus and Reliability Sharding Model in IoT Blockchain

Nianqi Jiang [1,2], Fenhua Bai [1,2], Lin Huang [3], Zhengyuan An [3] and Tao Shen [1,2,*]

[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650093, China; jiangnq@stu.kust.edu.cn (N.J.); bofenhua@stu.kust.edu.cn (F.B.)
[2] Yunnan Key Laboratory of Computer Technologies Application, Kunming University of Science and Technology, Kunming 650500, China
[3] Yunnan Provincial Academy of Science and Technology, Kunming 650051, China; yilinhl@sina.cn (L.H.); azylflin@163.com (Z.A.)
* Correspondence: shentao@kust.edu.cn

Abstract: The Internet of Things that links the cyber and physical worlds brings revolutionary changes to society, however, its security and efficiency problems have not been solved. The Consortium Blockchain + IoT is considered to be an effective solution. The IoT blockchain network's demand for transaction processing speed is gradually increasing. The throughput problem of the blockchain needs to be solved urgently and the security problem of transaction processing that comes with it. To solve the above problems, this paper proposes a reputation-driven dynamic node security sharding consensus model (RDSCM) in the blockchain, which consists of two parts: a reputation-driven node to eliminate PBFT (RE-PBFT) and a reputation-driven node cross reconfiguration sharding scheme (NCRS). The RE-PBFT eliminates abnormal nodes in the consensus network and reduces the probability of abnormal nodes becoming master nodes. NCRS improves the blockchain throughput while ensuring sharding reliability. Finally, the experiment proves that RE-PBFT can identify abnormal nodes and remove them in a short time. NCRS can effectively guarantee the reliability of sharding, and the transaction processing efficiency has been greatly improved after sharding.

Keywords: blockchain; Internet of Things; reputation; node eliminate; sharding

Citation: Jiang, N.; Bai, F.; Huang, L.; An, Z.; Shen, T. Reputation-Driven Dynamic Node Consensus and Reliability Sharding Model in IoT Blockchain. *Algorithms* **2022**, *15*, 28. https://doi.org/10.3390/a15020028

Academic Editors: Shiping Chen, Kaimeng Ding and Yanmei Zhang

Received: 16 December 2021
Accepted: 12 January 2022
Published: 18 January 2022

1. Introduction

The term "Internet of Things" [1] was first introduced in the ITU Internet Report 2005: Internet of Things, which was published by the ITU in 2005. It is essentially an intelligent network device capable of providing various industrial services [2]. It has become an indispensable technical help and data source to promote production [3]. Whether it is collecting information or identifying the authenticity of the device, it needs to be realized through communication between devices [4]. However, the security and efficiency problems caused by the heterogeneity of Internet of Things devices and centralized architecture have always existed [5]. Moreover, there are huge differences in computing resources and storage resources between devices, and devices with weak performance are vulnerable to malicious attacks [6]. Blockchain [7] has been developing rapidly since its birth. It is essentially a distributed database, which relies on an encryption algorithm, consensus algorithm, P2P network, and other technologies and links blocks together in a chain structure [8,9]. At present, blockchain has become an emerging technology with the same influence and prospects as big data, artificial intelligence, cloud computing, self-driving, and other technologies. Introducing new technologies to solve existing problems is an effective means [10].

The IoT blockchain [11] generated by the combination of blockchain and the Internet of Things can be effectively applied in the distributed network of IoT data sharing, user transactions, and other scenarios [12,13]. However, the throughput problem of blockchain,

which has been criticized for a long time, still exists. Bitcoin handles about 3–4 transactions per second, while Ethereum only handles 14 transactions per second [14]. Therefore, the IoT blockchain generally adopts the PBFT as the concordant protocol of the consortium blockchain. The consortium blockchain has the characteristics of openness, transparency, and decentralization, and its throughput is greatly improved compared with the public chain. Common consensus algorithms in the consortium blockchain include IBFT [15], PoA [16], PoET [17],, PBFT [18], etc., as shown in Table 1. PoA has a large delay and cannot meet the needs of the Internet of Things, while PoET mainly runs in Intel's SGX security environment and cannot be used in the Internet of Things environment with a wide range of devices; the theory and performance of IBFT and PBFT are almost the same, so the more classical PBFT is generally used as the common consensus algorithm in the Internet of Things blockchain, and PBFT also has some problems. With the development of IoT blockchain, the number of users participating in the consensus has increased, and the consortium blockchain has also exposed its problems. First, with the increase of users participating in the consensus, the transaction processing speed decreases; secondly, the increase in users may lead to the increase of abnormal nodes. Finally, the characteristic of PBFT is that it tolerates malicious nodes without corresponding measures, which may cause a distributed system to crash. Solving these problems is the focus of this paper.

Table 1. Comparison of consensus algorithms.

Name	Scalability	Throughput	Latency	IoT Suitability
PoET	high	high	low	no
PoA	high	low	high	no
IBFT	low	high	low	yes
PBFT	low	high	low	yes

To solve the above problems, this paper first proposes an evaluation scheme for quantifying node behavior. Based on the evaluation scheme, a reputation driven node elimination PBFT (RE-PBFT) is proposed to optimize PBFT consensus. Its advantages are as follows: (1) The master node is selected based on reputation. Only with high reputation can it become a master node and reduce the possibility that a malicious node becomes a master node and harm the consensus network; (2) The local outlier factor algorithm (LOF) is used to judge the abnormal nodes and remove them from the consensus network, to reduce the number of malicious nodes in the consensus network and make the total number of malicious nodes far away from the number of tolerated malicious nodes. In addition, a reputation-driven node cross reconstruction sharding scheme (NCRS) is proposed to ensure the reliability of sharding, and the concepts of ranking sharding and consensus sharding are proposed to improve the consensus speed after sharding. Its advantages are (1) improving the transaction processing speed of blockchain by dividing sorting sharding and consensus sharding; (2) NCRS ensures the reliability of segmentation. The contributions of this paper are as follows:

(1) To quantify node behavior, we propose a reputation evaluation scheme for node behavior, and design RDSCM based on this scheme;
(2) RE-PBFT is proposed, based on the evaluation scheme, including the primary node election scheme and abnormal node determination and elimination scheme, which not only reduces the abnormal nodes but also reduces the possibility of an abnormal node becoming the main node;
(3) The necessary conditions that sharding should meet are proposed and proved. Based on the evaluation scheme, NCRS is proposed to make the existing malicious nodes evenly distributed among all shardings and reduce the situation that the sharding is taken over by malicious nodes.

(4) The idea of sorted sharding and consensus sharding is proposed. Each consensus sharding produces a prepared block separately, which is processed by sorted sharding and linked to the blockchain.

2. Related Work

Related areas of work involved in RDSCM: 1. IoT blockchain; 2. Blockchain consensus; 3. Sharding technology.

2.1. IoT Blockchain

The combination of the Internet of Things and blockchain is getting closer and closer [19]. Technical application management schemes on the Internet of Things and blockchain data sharing have been launched to speed up technological innovation and application implementation. Nowadays, a blockchain is widely used in the Internet of Things [20,21]. In the literature of [22] is proposed a secure medical transaction access system using blockchain technology. By establishing two blockchains with different functions: personal medical (PHC) blockchain and external records (ERM) blockchain, it realizes blockchain transaction and secure access management, and accurately provides data to doctors under the condition of ensuring patient privacy and security and assists medical detection with machine learning. Two blockchains with different privacy levels can ensure the users' privacy, security and medical data sharing; The authors of [23] proposed the blockchain-based power transaction ecology (B-ET) in smart cities. By collecting intelligent instrument data in smart cities, transactions are delivered to a group of authorized nodes for verification, and then the blockchain is stored. Decentralization based on the blockchain can establish a safe and reliable power transaction system in smart cities. B-et proposed a PoW consensus mechanism based on reputation to overcome the problems of high delay of PoW and the lack of randomness of PoS, and finally ensure profit maximization through Stackelberg. In the literature of [24], a blockchain secure access control scheme is proposed, including an attribute-based factory control scheme and blockchain identity authentication scheme, to solve the problems of secure storage, access control, information deletion, and update of IIoT. The authors of [25] concluded that IoT and blockchain integration can produce a higher social economy and greater scientific research value. The authors of [26] use blockchain to realize the data security sharing of internal information, with the closer integration of blockchain and artificial intelligence technology [27]. In the future, new technologies are bound to be produced, combining IoT, blockchain and artificial intelligence.

2.2. Blockchain Consensus

Blockchain is essentially a distributed ledger. All users in the distributed network can record in the ledger through consistency protocols. There are many kinds of consistency protocols, and a consensus algorithm is the general name of these consistency protocols. Consensus algorithms can be roughly divided into two categories: proof class and voting class. The most famous proof consensus algorithm is the Proof of Work (PoW) mentioned by Satoshi Nakamoto in the Bitcoin White Paper [8]. In PoW, all users have the same bookkeeping right. The first user to calculate the PoW problem gets the bookkeeping right and will also get a certain amount of Bitcoin as a reward. PoW bookkeeping requires a large amount of computing power resources to carry out the worthless calculation, for which the Proof of Stake (POS) [28] is generated. In POS, the decision of the block producer is determined by the stake value mastered by the user. The higher the stake value, the more likely it is to become the billing node, but most of the stake resources are still controlled by a small number of users. The rich are getting richer. Different from the proof algorithms, voting algorithms do not require a lot of hashing and require a lot of communication between consensus participants, such as PBFT [18], RAFT [29], and Paxos [30], which all need to consume a large amount of communication resources. Both consensus algorithms are essentially ways for users to make themselves credible at a cost.

The consensus algorithm enables blockchain to play a great role in many scenarios. The authors of [31] believe that blockchain government will make 21 practical new forms of infrastructure. The authors of [32] believe that blockchain can produce a faster and more efficient tax invoice system in the tax field [33].

2.3. Sharding Technology

Sharding initially was a technology used in the database, then was introduced into a blockchain scalable plan, Its biggest advantage is that each subslice is parallel to the others, and different transactions are processed between shards. Moreover, for a consensus algorithm requiring a large amount of communication, the number of nodes after sharding is reduced, the communication times are reduced, and the performance of consensus is improved. Elastico [34] is the first blockchain project to adopt sharding technology. It adopts the method of PoW sharding +BFT consensus, which reduces the number of users of consensus and improves the transaction processing speed. A similar sharding scheme is also adopted in RapidChain [35], which randomly allocates shards according to the hash value of nodes. Both of the above two methods adopt PoW to deal with a certain part of sharding to some extent, so they both have the problem of resource waste caused by hash calculation. In addition, with the development of the consortium chain, the number of nodes participating in the consensus increases, and the sharding technology is gradually applied to the consortium chain from the public chain. Reference [36] proposed that consensus nodes could be divided into multiple consensus groups, and the transaction transmission between master nodes and consensus groups could be transferred by the benefits node, to achieve the purpose of sharding consensus in the consortium chain to improve consensus efficiency. Reference [37] proposed a multi-layer consensus mechanism, whose structure is similar to sharding, but sub-shards can continue to divide molecular shards and finally achieve a tree-shaped structure. This consensus realizes the way of infinite node division and maximizes the consensus speed of blockchain. In Reference [38], nodes in the consortium chain were divided into consensus domain nodes and storage domain nodes, and the consensus was divided into domains in the consensus domain to reduce the number of consensus nodes in a single domain, improve consensus efficiency, and separate storage and consensus to reduce the burden of storage/consensus.

3. System Model

The system model of RDSCM consists of two parts: a reputation-driven node elimination pbft (RE-PBFT) and a reputation-driven node cross reconstruction sharding scheme (NCRS). Figure 1 shows the structural relationship between RDSCM and IoT. The lower part is the IoT network. The data or transactions generated by RDSCM need to be sent to the blockchain network for identity identification. After the consensus and other processing, the blockchain is stored to complete the transaction.

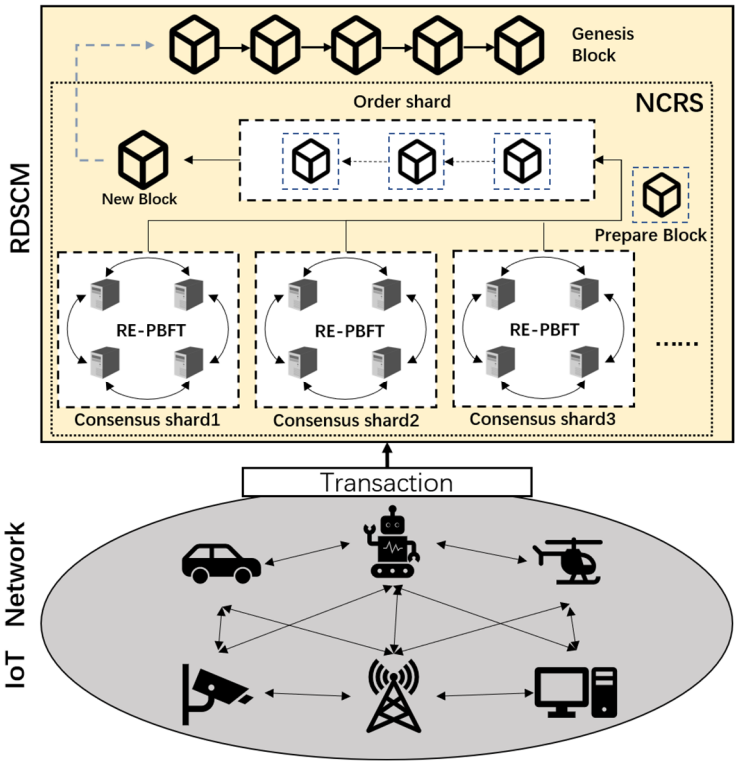

Figure 1. RDSCM applied in IoT network.

3.1. RE-PBFT

3.1.1. Reputation Introduction of RE-PBFT

The PBFT is a consensus agreement between nodes through multiple broadcast voting results. For the consensus process, under the condition of client security, the purpose of the normal working node is to verify and pass the transaction and then store it on the blockchain. However, in addition to the normal working node, there are also dishonest nodes and down nodes, which are respectively called Byzantine nodes and crash nodes. From a macro point of view, the three types of nodes in the consensus network will bring two different results: normal nodes are conducive to transaction verification and successful block generation; Byzantine nodes and crash nodes are the opposite. By analyzing the behaviors of the above three types of nodes in the consensus process, the following three behaviors can be obtained: normal consensus behavior, dishonest behavior, and node failure. dishonesty is detrimental to the consensus network, which includes both the malicious behaviors determined by Byzantine nodes themselves and the malicious behaviors caused by attacks on normal working nodes (such as man-in-the-middle attack). Node downtime may be due to the problem of the node itself (device performance problems or being attacked and unable to respond to requests), or it could also be that the Byzantine node acted against the consensus process. It is not easy to judge whether the node is dishonest from the behavior of the node. To solve the above problems, this paper proposes a reputation evaluation scheme to quantify node behavior. The basic idea of quantification is as follows: in the normal PBFT consensus algorithm, the number of normal working nodes is greater than the number of abnormal nodes, and the system can only be effective if this requirement is met. Therefore, the nodes with most behaviors are defined as normal working nodes, and the nodes with few behaviors are defined as abnormal nodes. The normal working nodes are rewarded with reputation, and the abnormal nodes are punished. Section 3.1.3 introduces the judgment basis of abnormal nodes based on reputation.

Reputation is the credibility of a node. The higher the reputation, the higher the possibility that the node is a normal working node, and the higher the possibility that the node is a normal working node in the consensus process (the stronger the ability to resist

attacks and system collapse), and for the smaller, the opposite is true. The reputation is obtained by quantifying the behavior of nodes. The quantification standard is based on the influence of the behavior of nodes in the consensus network. If the influence is large, the change of reputation will be correspondingly larger.

Based on the above ideas, the following reputation replacement scheme is proposed:

For normal nodes:

$$r_t = \begin{cases} r_{t-1} + 1 + v_i, node\ normal \\ r_{t-1} - p^{n-1} - (total - 1)^2 - v_i, node\ abnormal \end{cases} \quad (1)$$

For main nodes:

$$r_t = \begin{cases} r_{t-1} + 1 + v_i, node\ normal \\ r_{t-1}/2 - p^{n-1} - (total - 1)^2 - v_i, node\ abnormal \end{cases} \quad (2)$$

where r is the reputation value of the node; t is the number of consensus, n is the number of successive abnormal behaviors of this node; p is the continuous abnormal punishment factor, generally take the natural constant e; total is the cumulative number of abnormal; $v_i = 1 - (i-1)/N$ represents the reward/penalty value of the ith node that completes the consensus, where N is the number of nodes participating in the consensus in the same shard, v_i is the additional reward in the case of normal working nodes, otherwise, it is extra punishment. Nodes with downtime behavior will be punished with a random value in (0, 1].

The normal operation of nodes is the obligation of all nodes, so the reputation will increase the baseline value after the normal operation of nodes. To motivate nodes to complete the consensus more quickly, there will be v_i of different sizes according to the order of nodes to complete the consensus, and v_i can be calculated by the arithmetic sequence general term formula; when the node is abnormal, the normal node exception will only affect the voting situation in three-stage communication, and the influence on the system is small when it does not exceed the maximum limit that PBFT can tolerate. However, when the abnormal node becomes the master node, it may cause the failure of the whole consensus process, a more severe reputation penalty mechanism is needed to halve the reputation of the node that is abnormal in the master node. Whether it is the master node or the ordinary node, the abnormal behavior will not only be punished according to the total number of exceptions, but also severely punished for the continuous number of exceptions, and the reward for completing the consensus will become the corresponding size of punishment. It can be seen that when the node is the first abnormal, the number of consecutive exceptions and the cumulative number of exceptions are both 1, so the penalty formula is equal to $r_t = r_{t-1} - 1 - v$ or $r_t = r_{t-1}/2 - 1 - v_i$ (the reputation equivalent formulas for normal node and master node, respectively), punishment is as strong as incentive; the nodes with continuous abnormal behaviors are more likely to be abnormal nodes, and the punishment is more severe than the cumulative abnormal behaviors. If there is a node with the reputation $r_i \geq 50$, there is a scaling ratio $\delta = 25/r_i$, and the reputation of all nodes is multiplied by the scaling ratio δ to prevent the reputation from increasing unrestrictedly.

Each node in the distributed network maintains a node reputation table locally just as it maintains the blockchain, as shown in Figure 2. After the completion of the consensus round, the node calls the smart contract [39] calculated with a reputation based on the consensus results received from the other nodes. The new reputation values of all nodes are calculated. It is then sent to all the remaining consensus nodes; when the consensus node receives the new reputation sent from the other nodes, it saves the reputation in the local temporary table. When the same result reaches f + 1 (f is the maximum number of malicious nodes that can be tolerated), the consensus node saves the final reputation value locally.

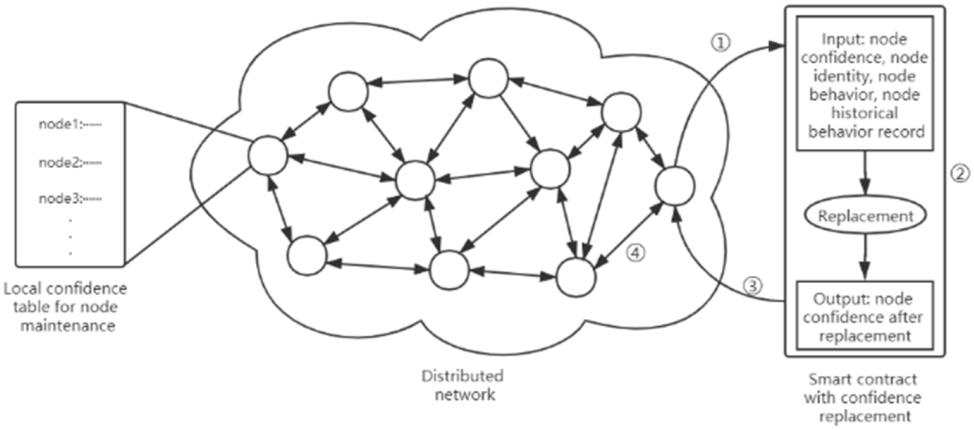

Figure 2. Reputation replacement based on smart contract.

In Figure 2, (1) indicates that the node passes the local reputation related parameters into the smart contract to update the reputation; (2) represents the smart contract invocation process; (3) indicates that the calculated reputation is returned to the node; (4) sends the reputation value calculated by nodes to other nodes or receives the result value from other nodes.

3.1.2. Master Node Election of RE-PBFT

To solve the problem that the traditional PBFT master node rotation election may lead to malicious nodes becoming the master node, in the consensus network, nodes with high reputation are relatively more reliable and have a higher possibility of working normally. Therefore, the top 50% of nodes with reputation are selected as the master node preliminary interval. To increase the randomness, to prevent the occurrence of the same node becoming the primary node continuously, 60% of the nodes in the preliminary interval are selected as the master node interval by random probability, the master node is selected from the master node interval. The node that became the master node in the last round can only appear in the preliminary interval in the next round and cannot enter the master node interval again. To increase the randomness and unpredictability of principal nodes, we decided to select principal nodes through VRF (verifiable random function).

Verifiable random function [40] is a pseudo-random function proposed by Micali et al. in 1999. It consists of a key generation function, random number generation function, proof value generation function, and verification function.

- Key generation function: RSA digital signature algorithm is used to generate a pair of public and private keys, the public key is PK, the private key is SK;
- The random number generating function: The random number β is obtained from the given input and the private key of the node through Equation (3), where *Vrf_hash*() is the hash function and α is the input.

$$\beta = Vrf\_hash(sk, \alpha) \tag{3}$$

- Proof value-generating function: Prove β is the correct output of the input α. First, obtain the zero-knowledge proof result π of the input α through Equation (4). Then, any node can obtain whether β is the correct output by taking π as the input of formula (4).

$$\pi = Vrf\_prove(sk, \alpha) \tag{4}$$

$$\beta = Vrf\_prove\_to\_hash(\pi) \tag{5}$$

- Verification function: After Equation (5) proves that the output β is obtained from the input α, the result of Equation (6) is calculated, True means that the verification is passed, while False means that the verification is not passed.

$$Vrf\_verify(pk, \alpha, \pi) \tag{6}$$

The proportion of reputation of each node in the master node interval is calculated by the Equation (7). m is the number of nodes in the master node interval, r_i is the reputation of node i, and p_i is the proportion of reputation of a node i in the master node interval. The probability space of the master node can be obtained $B_i = \left[\sum_{j=0}^{i-1} p_j, \sum_{j=0}^{i} p_j\right)$ nodes.

$$p_i = \frac{r_i}{\sum_{j=1}^{m} r_j} \tag{7}$$

First, the public key PK and private key SK are generated. Then, given the random input seed number α, β_i is obtained by Equation (3). The result bits of the hash are consistent, β_{len} is the data length of β_i. The zero-knowledge proof result is calculated by the Equation (4), and then the node sends the β_i and π obtained by local calculation to the other consensus nodes. After receiving the data from other consensus nodes, calculate Equation (5). If it is correct, calculate β_i/β_{len}. Count the number falling in each β_i interval, and the one with the largest number will be the main node. Finally, Equation (6) is used to verify whether all the results are calculated α.

3.1.3. Node Eliminate of RE-PBFT

Nodes in distributed networks are peer to peer, and there is no authoritative central department to verify the reliability of nodes. Although the behavior information of a node can be obtained through the reputation of a node, there is a lack of clear criteria for judging whether it is an abnormal node. To try to remove a node, it is necessary to judge whether the node is abnormal first. In RE-PBFT, the local outlier factor algorithm (LOF) in anomaly detection is used to test the abnormal node. In reference [41], the LOF introduces a local outlier value to each object in the data set to represent its outlier degree. This is the first concept of outliers, which quantifies how outlier an object is. Outlier factors are local, only one restricted neighborhood of each object is considered, and the method is related to the sample density in the neighborhood of the object.

The sample density is determined by the range of the neighborhood and the number of samples in the neighborhood. There is a parameter K in LOF to determine the size of the neighborhood of the object, K is the number of samples around the object, and the circle with the radius from the object to the kth far sample is the kth field of the object. The selection of the K value determines the judgment accuracy of outliers, the K value is set to 2/3 in this model because it has been known in the PBFT consensus algorithm that the proportion of maximum malicious nodes is 1/3 and that of normal working nodes is 2/3. It cannot directly contribute to the reputation as an LOF of input, one dimensional feature is too simple to obtain good results, so the two-dimensional data is composed of the reputation of the end of the previous round of consensus and the reputation of the end of the current round of consensus, expressed as $R_i = \left(r_i^{t-1}, r_i^t\right)$, The above formulas respectively represent the reputation of node i after the $t-1$st consensus and the reputation of node i after the tth consensus, R represents the input data of node i in LOF. Based on LOF, the threshold value of the outlier factor is designed, which is the average value of the outlier factor of all nodes. The threshold value is adopted because the outlier factor will change greatly after one abnormal behavior exists in the node. However, as described in Section 3.1.1, the abnormal nodes are judged under the accumulation of abnormal behaviors, so the threshold value of the outlier factor is compared with the outlier factor value of each node to determine whether any node is removed.

The specific implementation process is shown in Figure 3, and the implementation method is as follows:

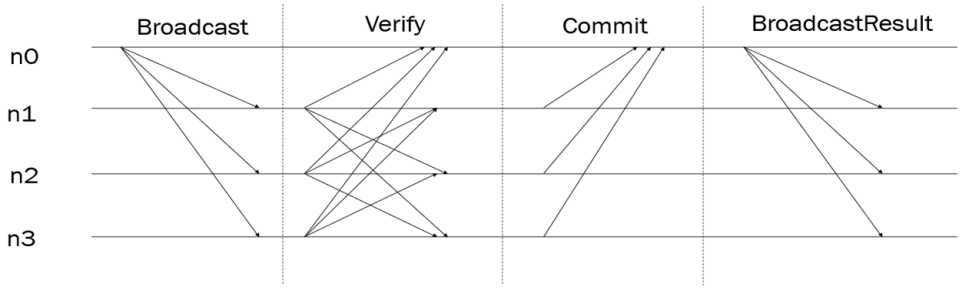

Figure 3. Node elimination consensus process.

Step 1: After the consensus is completed, the node reputation is updated by Equations (1) and (2) according to the behavior of the node;

Step 2: The node checks and calculates the local maintenance reputation table. If it finds any node that meets the condition of being removed, it will pack the discovery as <<*Broadcast*>, *message*, *timestamp*, *sign*> and rebroadcast to all remaining nodes (<*broadcast*> represents the sending stage, where the message is the message body, the *timestamp* is the sending timestamp, and *sign* is the signature of the sending node);

Step 3: Node receives node eliminate broadcast from other different nodes <<*Broadcast*>, *message*, *timestamp*, *i*>, if the messages are the same, the *timestamp* order shall be taken as the standard, and the first broadcast node shall be the discovery node eliminated this time. If different, it will be processed in the order of *timestamp*;

Step 4: Verify by comparing the message in <*Broadcast*> with the locally maintained reputation table. The verification result is packaged into <<*Verify*>, *verifyResult*, *sign*> and broadcast to all remaining nodes (<*Verify*> is the verification phase, *verifyResult* is the verification result);

Step 5: Node receives from other nodes <<*Verify*>, *verifyResult*, *sign*> information, count the number of nodes received, and count <*Verify*> in the *verifyResult*, receiving $2f$ of the remaining nodes <*Verify*> after the information (plus itself is $2f + 1$), the *verifyResult* which is more in the statistics is packaged as <<*Commit*>, *verifyResult*, *sign*> send to the discovery node (<*Commit*> represents the submission phase, *verifyResult* represents the verification result after statistics);

Step 6: After the discovery node receives f <<*Commit*>, *verifyResult*, *sign*> from the other different nodes (plus itself is $f + 1$). Then the eliminate information result <<*BroadcastResult*>, *result*, *sign*> of this node can be broadcast to all other nodes (<*BroadcastResult*> is the result broadcast stage, and result is the final result);

Step 7: The remaining nodes receive the eliminated result broadcast from the discovery node <<*BroadcastResult*>, *result*, *sign*>, the eliminated node will be removed from the P2P network and no longer communicate with the node.

3.2. Node Cross Reconfiguration Sharding Scheme (NCRS)

Blockchain sharding is a common method of expansion on the chain. To solve the possible sharding failure problem, a blockchain sharding distribution scheme NCRS is designed to achieve reliable node sharding. After sharding, order sharding and consensus sharding are adopted. Blockchain sharding is to improve the transaction throughput of the blockchain, and the NCRS scheme aims to minimize the possibility of sharding failure. The sharding scheme implementation is described in detail below.

3.2.1. Limited Number of Shards

The number of shards is the premise to ensure the effectiveness of sharding, and an improper number of shards will lead to the failure to achieve the safety and effectiveness of each sharding no matter how the nodes are divided. It is assumed that there are N nodes in the blockchain, where the proportion of malicious nodes is $\delta(\delta \leq 1/3)$, the number of

shards is k and Ni is the number of the ith shard nodes. The number of shards must meet Lemma 1.

Lemma 1. *The number of shards must meet* $k \leq min\{N(1-3\delta),\ N/4\}$ *to ensure the safety of all shards.*

Proof: The minimum consensus size in PBFT requires the existence of four nodes, so the maximum number of shards cannot exceed *N/4*. After sharding, each shard runs separately PBFT internal consensus, so the internal shard can tolerate the maximum number of malicious nodes is $\lfloor (N_i-1)/3 \rfloor$, and the sum of the number of malicious nodes that can be tolerated by each shard must not be less than the number of malicious nodes that can be tolerated before the sharding, so there will be $\sum_{i=1}^{k} \lfloor (N_i-1)/3 \rfloor \geq \lfloor N\delta \rfloor$, through the zooming method can get $\sum_{i=1}^{k} (N_i-1)/3 \geq N\delta$, Substitute $\sum_{i=1}^{k} N_i = N$ into $\sum_{i=1}^{k} (N_i-1)/3 \geq N\delta$ and you obtain $k \leq N(1-3\delta)$. Therefore, to ensure the reliability of sharding, the number of shards must meet $k \leq min\{N(1-3\delta), N/4\}$. □

Although the reliable sharding scheme must satisfy Lemma 1, the reliable sharding scheme may not be obtained by satisfying Lemma 1. Lemma 1 is only a necessary and insufficient condition for the reliability of the sharding scheme.

3.2.2. Details of NCRS

Sharding is an effective solution to improve the throughput of blockchain. The core requirement of sharding is randomness during sharding and reliability after sharding. To prevent excessive malicious nodes appearing in the same shard leading to the malicious node taking over, this paper shows the design of a shard distribution scheme based on the reputation calculation method introduced in Section 3.1: the minimum value of the sum of the absolute values of the difference between the mean reputation of each shard and the mean reputation of all nodes is obtained. In other words, the sharding scheme satisfying Equation (8) is obtained. Equation (8) is defined as the fitness function, k is the number of shards, N is the total number of blockchain nodes, N_i is the number of nodes in the ith shard, and r is the node's reputation.

$$Fit = min \sum_{i=1}^{k} \left| \frac{\sum_{j=1}^{N_i} r_j}{N_i} - \frac{\sum_{j=1}^{N} r_j}{N} \right| \tag{8}$$

However, there are two problems in solving the fitness function: (1) The fitness function is an NP-hard problem, and the result cannot be calculated in polynomial time. (2) Even if the optimal solution of the fitness function can be solved, there is still a problem: the optimal solution is unique or predictable, which cannot meet the randomness requirement of sharding. To solve the above problems, in this paper, we designed the Node cross Reconfiguration Sharding (NCRS) scheme of blockchain nodes based on reputation.

The total number of shard schemes can be calculated according to the combination number formula shown in Equation (9), where N is the total number of nodes and M is the number of nodes in a single shard. In the actual calculation process of the fitness function, in addition to the optimal solution, other sharding schemes can also achieve reliable shards. Therefore, the NCRS scheme uses the iterative method to calculate the fitness function. In the iterative process, the node cross reconstruction sharding method is used to obtain a reliable sharding scheme, it does not need to calculate the fitness function solution of the final, it just needs to obtain a certain number of iterations that can be regarded as the result of the reliable shard met sharding scheme. The problem of time consumption and loss of randomness in calculating the optimal solution is solved. Using the total result calculated by the combination number formula as the benchmark, it is found through

several experiments that the success rate of sharding with 100% reliability can be achieved when the number of iterations reaches 70% $c(n, m)$.

$$c(n, m) = \frac{n!}{(n-m)!m!} \tag{9}$$

The implementation process of NCRS is shown in Algorithm 1. The external main body is iteration, and the internal is node cross reconstruction sharding. The number of alternative sharding schemes is quite large. To obtain a successful sharding scheme in a limited number of iterations, targeted sharding reconstruction is required, The new sharding scheme is reconstructed by the method of cross nodes—randomly select a node higher than the average reputation of all nodes in the sharding with the highest average reputation and randomly select a node lower than the average reputation of all nodes in the sharding with the lowest average value for exchange. After the exchange, the average values of the new reputation of the highest and lowest pieces of pre-reputation are close to the average value of the overall reputation, which reduces the value of the fitness function to a certain extent. There are two slices in Figure 4. The reputation of four nodes in slice S1 is [1–4], and the reputation of four nodes in slice S2 is [5–8]. The fitness function value before crossover is calculated to be 4. After crossing, you can see that the node with the reputation of 8 in S2 is exchanged with the node with the reputation of 3 in S1. At this time, the reputation of S1 is [1,2,4,8], and the reputation of S2 is [3,5–7], and the fitness function value after the crossing is calculated to be 2.25. The purposeful node crossing between shardings can help NCRs gradually find the optimal sharding distribution scheme satisfying the fitness function.

Algorithm 1: Node cross reconstruction sharding scheme based on reputation.

Input: Number of Iterations Count,
Initialization: Random Shard, $Fit = +\infty$, Current Iterations CL = 0;
While $CL < Count$:
Then CL++
Crossover the shards, get a new shard distribution newShard;
Calculate the fitness value newRes;
If $newRes \leq Fit$ then
$Fit \leftarrow newRes$
$Shard \leftarrow newShard$
End
Output Shard

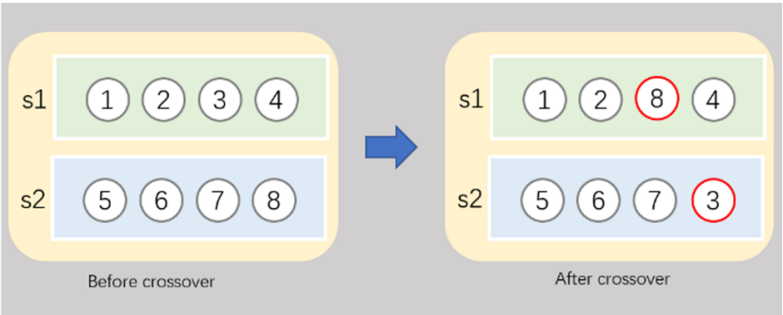

Figure 4. Schematic diagram of node crossing between patches.

After a reliable sharding scheme is obtained, the sharding is randomly divided into two categories: order sharding and consensus sharding. There is only one order sharding, and all other shardings are consensus sharding. The function of consensus sharding is to achieve consensus in the acquired transactions, package them into prepared blocks and send them to the order shard. The functions of order shards are: (1) to obtain transactions

and forward them to consensus shards; (2) package the prepared blocks to generate a complete block. The consensus process between shards is shown in Figure 5.

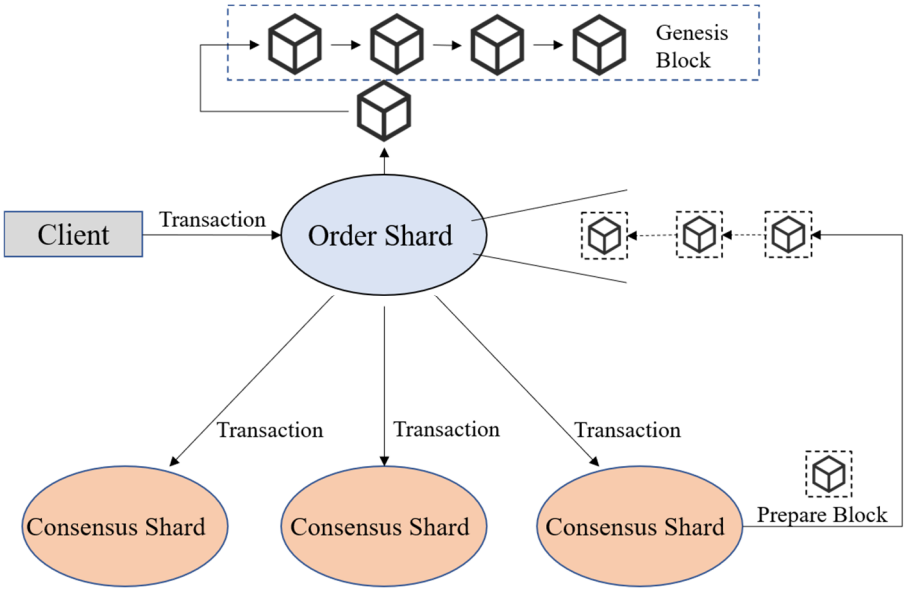

Figure 5. Relationship between shards.

After sharding, the order shard communicates with the client and obtains the transactions sent by the client, and sending the transactions of the same user to the same common sharding for consensus according to the signature of the users participating in the transaction, to realize transaction sharding and reduce the communication volume between the shards; The transaction only completes the consensus within the consensus shard. After completion, it is packaged into a prepared block and sent to the order shard. The order shard processes the prepared block into a complete block for transmission and broadcasting (there is no parent hash in the prepared block because the parent block that needs to be linked cannot be determined when the prepared block is generated, and the parent block can be determined only after order block processing); the processing order of the blocks in the ordering sharding shall be subject to the timestamp of the block received by the order sharding.

Sharding is dynamic and not fixed. It needs to be rebuilt in the following two cases: (1) when nodes are eliminated; (2) after completing an epoch consensus. Obtain the reputation environment of the current node, build a new sharding through NCRS, and make the sharding unpredictable by dynamic sharding.

4. Experiment and Result Analysis

4.1. Experimental Environment

The experimental environment was built by Java language, the remote call was realized by GRPC, transaction sorting was carried out by Redis as a queue tool, and data persistence was carried out by MySQL. The physical environment platform in which the experiment was run was an Intel Xeon Processor with a frequency of 2.4 GHz. The OpenStack cloud platform environment was deployed on this physical environment, and the specific configuration is shown in Table 2. The database and Redis ran separately in the Docker environment.

Table 2. Experimental environment parameter configuration.

Name	Parameter	Note
CPU	2.4 GHz	Single-core and single thread
RAM	2 GB	DDR4
ROM	20 GB	Mechanical drive 7200 r/min

4.2. Reputation Replacement Test

Reputation is the most basic idea in this paper and is the guarantee of the reliability of master node election, the mathematical basis of node elimination, and the theoretical core of reliable sharding. Ten rounds of consensus reputation reward and punishment tests were carried out on 10 nodes. Node 1,2,3,4,5,6,8 is the normal working node, node 10 is the continuous malicious node, node 9 is the discontinuous malicious node, node 7 is the malicious node when it becomes the main node. The change of node reputation is shown in Figure 6. It is observed that the reputation of the No. 10 node decreased after it acted maliciously in the first round of consensus, and it was judged as an abnormal node by the LOF algorithm and removed. Node 9 did not act maliciously continuously, and it worked normally before it acted maliciously. It can be seen that the reputation increased first and then decreased. Although the reputation remained around 25 after the second round of consensus, there was a big gap between it and the normal working node, and it was judged as an abnormal node by the LOF algorithm. The first four rounds of consensus of Node 7 all worked normally, and the reputation gradually increased. When the fifth round of consensus became the main node, Node 7 began to behave maliciously, reputation decreased considerably, and it was judged as an abnormal node by the LOF algorithm. All the rest of the nodes were slowly building reputation according to the rules.

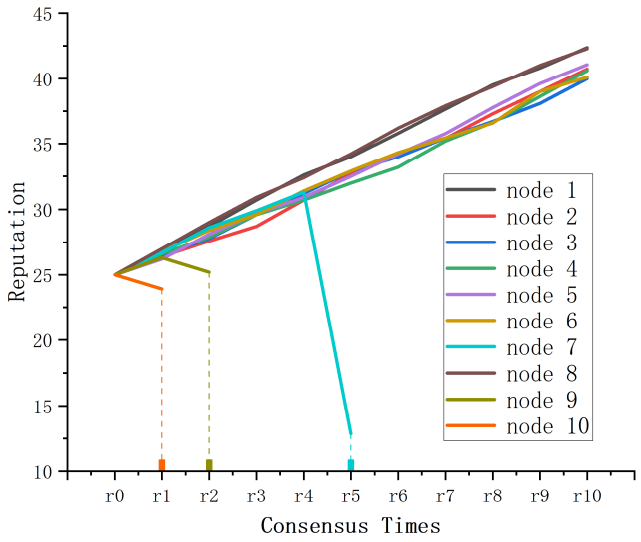

Figure 6. Changes of reputation of 10 nodes.

4.3. Comparison of the Average Reputation of Each Shard

To test the reputation distribution after sharding, without considering the node under the malicious condition, 20 normal working nodes built into three different subdivisions, for convenience for the experiment, each consensus was rebuilt after the completion of a shard. As observed in Figure 7, the shards' average reputation changed, each subdivision of the average reputation was on the rise, and the difference in value between the shards was very small, In general, it was the same as the total average reputation of all nodes and met the requirements of the fitness function.

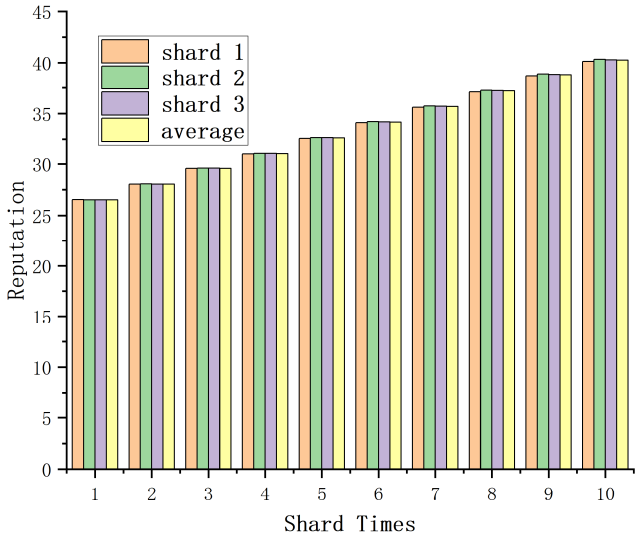

Figure 7. Comparison of reputation of different shards.

4.4. Local Outlier Factor (LOF) Experiment

To test the reputation-based abnormal-node judgment, the following five experimental situations were designed: (1) There is one continuous malicious node in 10 nodes; (2) There are two consecutive malicious nodes in 10 nodes; (3) There are three consecutive malicious nodes in 10 nodes; (4) There are three probable malicious nodes in 10 nodes; (5) In 10 nodes, there is one node that acts maliciously when it becomes a master node, one node that is continuously malicious, and one node likely to be malicious.

After each round of consensus, the updated reputation was calculated by the LOF algorithm, and the test results of the above five cases were visualized as the situation shown in Figure 8 below. In the figure, the horizontal axis is the reputation value after the previous consensus, the vertical axis is the reputation value after the current consensus, the red dotted line is the threshold curve, and the nodes outside the red range are judged as abnormal nodes, which need to be removed.

Figure 8A is the result of case (1). After the first round of consensus, nodes with malicious behaviors were judged as abnormal nodes. Figure 8B,C are the results after the first two rounds of consensus. Since there were two nodes with malicious behaviors, the reputation of both nodes decreased after the first round, which had a great impact on the value of the outlier factor in the LOF algorithm, so it was not considered an anomaly in Figure 8B. After that, two consecutive malicious behaviors led to a large reputation gap with normal nodes, which was judged as an abnormal node in Figure 8C. Figure 8D–F are the results of case (3) after the first three rounds of consensus. It can be seen that, in Figure 8F, the abnormal nodes are determined. It can be seen that the more abnormal the nodes, the more consensus time is needed to judge the nodes with malicious behavior, but the abnormal nodes can still be determined quickly by the continuous malicious behavior.

Figure 8G–J are the results of the second, fourth, sixth, and eighth round consensus in case (4), respectively. From the above analysis, it can be seen that the malicious behavior of a single node is easy to judge as abnormal, so the three nodes are defined as acting indirectly maliciously at the same time. From Figure 8G–I, it can be seen that the three nodes with malicious behavior are gradually farther away from the normal node cluster and closer to the threshold line, and the nodes in Figure 8J are judged as abnormal nodes. Even if it is not consecutive malicious behavior, the indirect malicious nodes can be judged as abnormal nodes within a small number of consensus rounds.

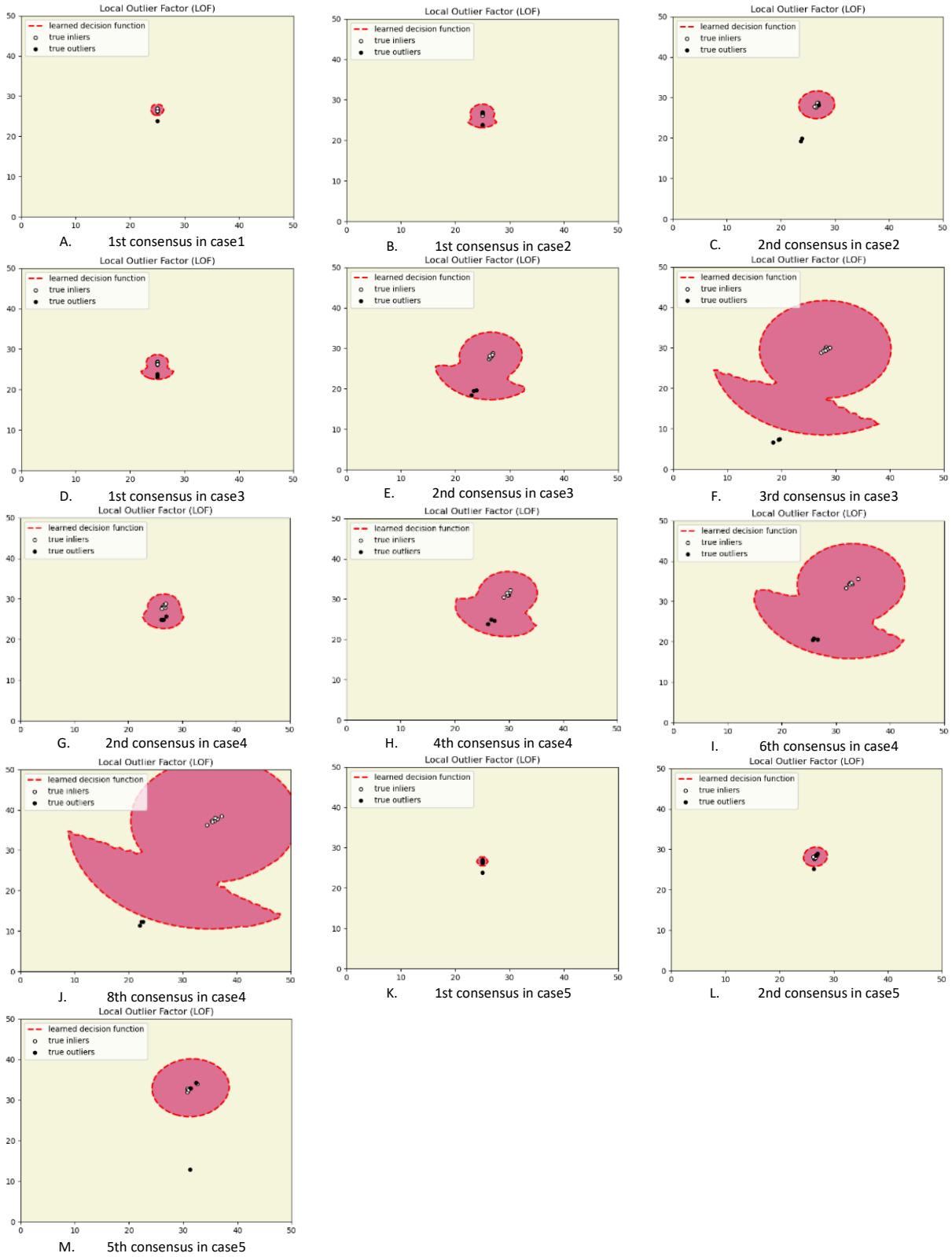

Figure 8. LOF experimental results. (**A**) is the result of case (1), (**B**,**C**) are the result of case (2), (**D**–**F**) are the result of case (3), (**G**–**J**) are the result of case (4), (**K**–**M**) are the result of case (5).

Figure 8K,L are the results of the first, second, and fifth consensus, respectively, in case (5). The node reputation of case (5) is as shown in Figure 6. In the first round of consensus, the probabilistic malicious nodes worked normally, and only one node was

judged as an abnormal node for malicious behavior. In the second round of consensus, there was only one node in the consensus network that acted maliciously and was judged as an abnormal node. In the fifth round, when the master node acted maliciously, the point became the master node, started to act maliciously, and was judged as an abnormal node. Since the master node is punished more severely for malicious behavior, it can still be judged as an abnormal node quickly even though the master node has a high degree of reputation before becoming malicious.

Table 3 shows the results of the comparison with the other three papers on node elimination schemes. In Reference [42], the author adds the concept of participation degree (PD). The upper limit of PD value is 3. For each consensus completed, PD is added by one, otherwise, it is reduced by 1. When PD is 0, it is eliminated. When a node does evil indirectly, it cannot be identified as a malicious node for elimination; The health status is introduced in Reference [43]. Each node monitors the health status of other nodes. If the node participates in the consensus, the health status is increased by 1 (it cannot exceed the maximum health value), otherwise, it is reduced by 1. When it is 0, it is deleted. The disadvantages are the same as in the previous article. Nodes with indirect malicious behavior cannot be eliminated; In Reference [44], the authors judge whether nodes need to be eliminated by voting, which would increase the communication consumption of the system and may be used by malicious nodes to endanger the network environment; the scheme in this paper can identify nodes as malicious nodes and eliminate them in a variety of evil situations, and will not waste communication resources.

Table 3. Comparison of node elimination schemes.

Scheme	Disadvantage	Advantage
Participation degree [41]	The participation table needs to be maintained locally, which consumes storage space; Unable to handle indirect malicious behavior nodes;	Continuous evil nodes can be quickly determined;
Health status [42]	The participation table needs to be maintained locally, which consumes storage space; Unable to handle indirect malicious behavior nodes;	Continuous evil nodes can be quickly determined;
Vote [43]	It consumes a lot of communication resources and may be used by malicious nodes, resulting in network congestion;	Not need to store other data, saving storage resources; It can accurately identify malicious nodes;
RE-PBFT	The participation table needs to be maintained locally, which consumes storage space;	It can solve the problem of evil behavior in different scenarios, and vote only when it is determined that there are nodes that need to be eliminated, which will not cause waste of communication resources;

4.5. Reputation Sharding Test Results

In Figure 9, with the increase of the proportion of malicious nodes in the total blockchain network under different sharding numbers, it can be seen that the three broken lines of the NCRS sharding construction scheme had a sharding success rate of nearly 100% when the proportion of malicious nodes was less than 0.25. After that, the larger the number of sharding, the lower the success rate; PBFT can only tolerate 1/3 of the number of malicious nodes at most. Therefore, when the proportion of malicious nodes reaches

33%, the sharding success rate is zero, but it has significant advantages compared with the sharding success rate of multi-layer PBFT [36].

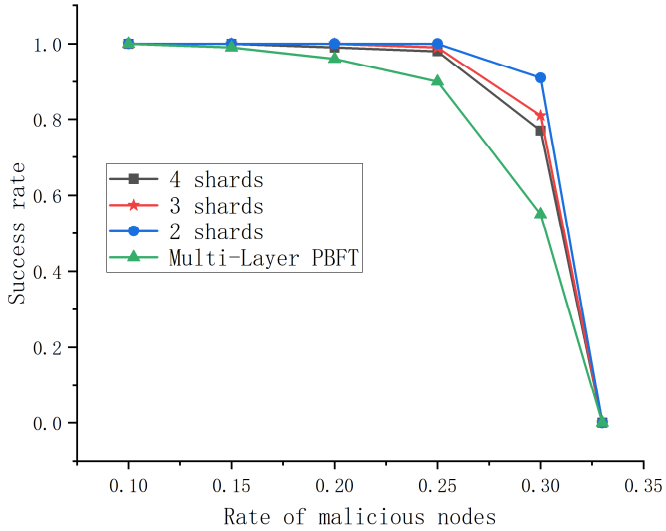

Figure 9. NCRS shard success rate.

4.6. Throughput Test for RDSCM

Throughput (TPS) is the number of transactions a blockchain can process per second and is an important measure of blockchain performance. Consensus speed is one of the important influencing factors of blockchain throughput (TPS). There is a positive correlation between consensus speed and TPS. The general formula for calculating TPS is as follows:

$$TPS = \frac{transSum}{\Delta t} \tag{10}$$

The above Equation (10) *transSum* refers to the number of transactions processed and Δt refers to the time spent processing transactions.

The TPS performance of the PBFT algorithm, hybrid algorithm [45], POC algorithm [46], and RDSCM in this paper are tested under different node numbers. The results are shown in Figure 10. It was found that the throughput of RDSCM was significantly higher than that of other algorithms under different node numbers.

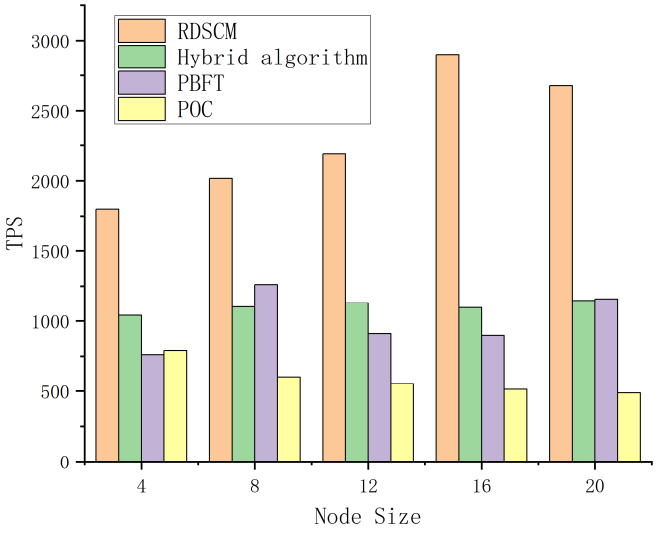

Figure 10. TPS comparison under different numbers of nodes.

Figure 11 shows the TPS comparison of RDSCM under the different numbers of shards with 16 nodes. It can be observed that TPS is very low when the number of shards is not divided, and the increase of consensus shards can bring a significant increase in TPS. It was also found that TPS is related to the number of transactions stored in a single block. According to Figure 12, although the increase in the number of transactions stored in a block increases the processing time correspondingly, the TPS calculation formula found that TPS would eventually increase with the increase in the number of transactions. It can be seen in Figure 13 that the higher the number of shards under the different numbers of shards, the less time consumption. In addition to the time consumed by consensus, the RDSCM also adds the time consumed by the confidence calculation and LOF calculation. As sees in Figure 14, the time consumed by these two parts is very small, accounting for only 2.4% of the time consumed by consensus, compared to Figure 11.

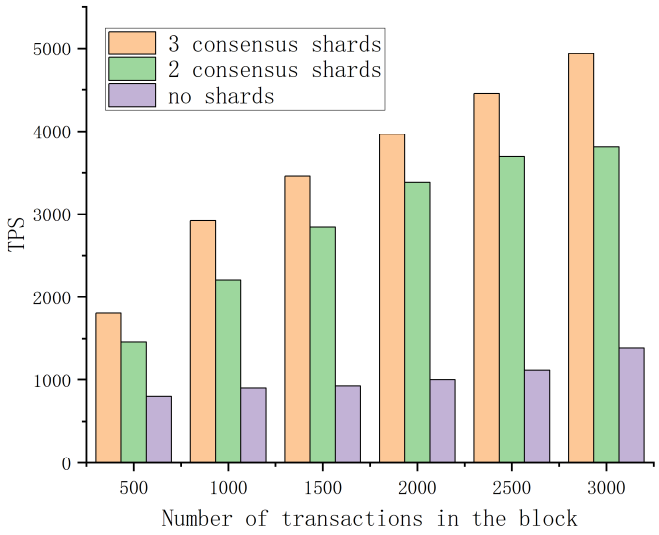

Figure 11. TPS comparison under different shard numbers.

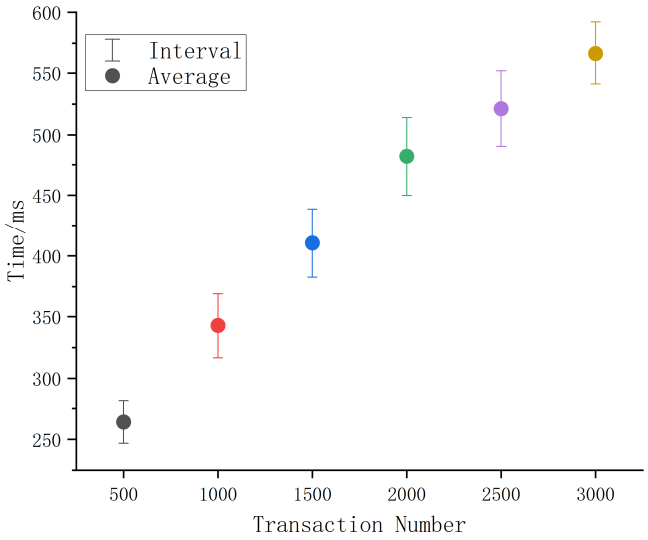

Figure 12. Consensus time consumption under four shards.

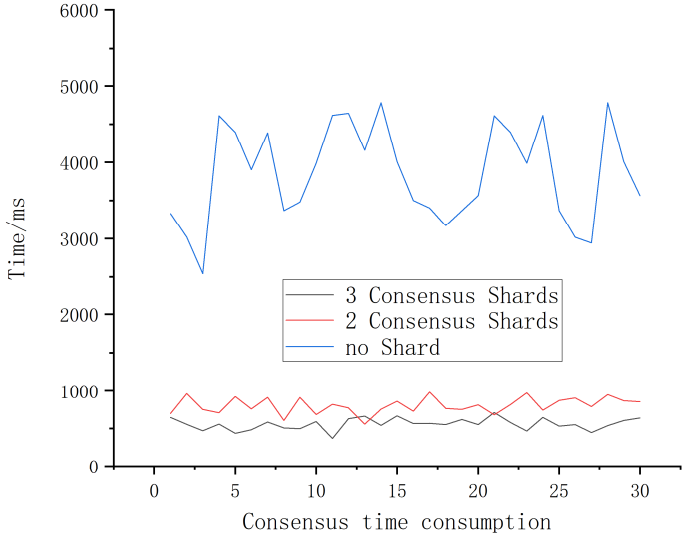

Figure 13. Consensus time consumption.

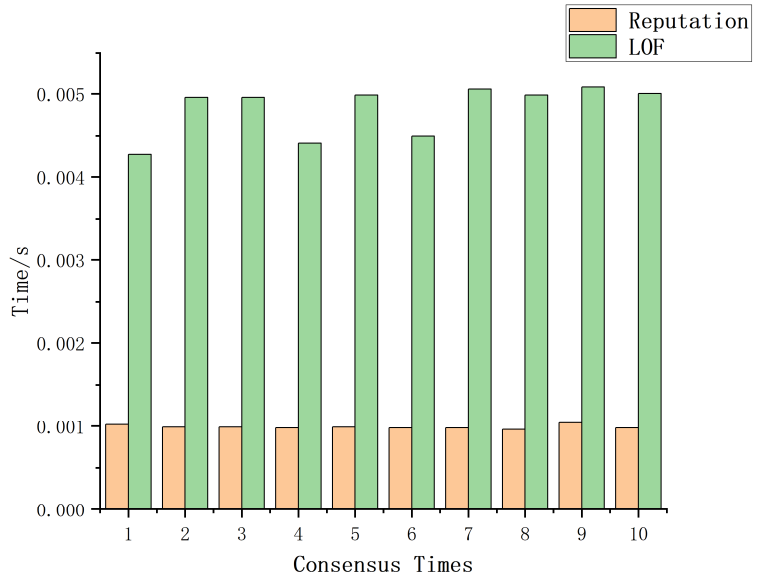

Figure 14. Reputation and time consumption.

5. Conclusions

In this paper, an RDSCM blockchain scheme is proposed. By introducing the concept of reputation and combining VRF and LOF algorithms, RE-PBFT is proposed to reduce the number of abnormal nodes and the probability of abnormal nodes becoming the main node. Then, the NCRS algorithm is proposed to ensure the shard is reliable to realize the capacity expansion of the blockchain, and a new consensus structure after sharding is proposed. RDSCM improves the security and throughput of the blockchain. The experimental results show that the NCRS scheme can minimize the difference between the mean reputation of each sharding, and has a high success rate of sharding in the case of complete node sharding, The experimental results show that the success rate of the sharding is high, and it also has significant advantages compared with the existing articles. The RE-PBFT of any abnormal nodes can be eliminated in less consensus time, and the fewer nodes having abnormal behaviors at the same time, the easier it is to eliminate them. The instances of evil behavior that can be handled are richer than those eliminated in other papers. Finally, the TPS experiment found that there were considerable changes before and after sharding, and the effect of the increased reputation and LOF calculation on consensus was

almost negligible in the RE-PBFT. Compared with other consensus algorithms, the TPS performance of RDSCM has been greatly improved.

6. Future Work

This paper has shown fast efficiency in reputation value calculation and node elimination. However, to increase the success rate of segmentation, it would take a long time to calculate the segmentation scheme through iteration. In the future, it would be necessary to further improve the calculation efficiency when calculating the segmentation scheme.

Author Contributions: Conceptualization, N.J.; methodology, N.J.; software, N.J.; validation, N.J., L.H. and T.S.; formal analysis, N.J.; investigation, N.J.; resources, N.J.; data curation, N.J.; writing—original draft preparation, N.J.; writing—review and editing, N.J., F.B. and T.S.; visualization, N.J.; supervision, F.B. and T.S.; project administration, T.S.; funding acquisition, Z.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank the Major Scientific and Technological Projects in Yunnan Province (no. 202002AB080001-8),Yunnan Key Laboratory of Blockchain Application Technology (funding no. 202105AG070005, open project no.YNB202109), the National Natural Science Foundation of China (No. 61971208), Yunnan Reserve Talents of Young and Middle-aged Academic and Technical Leaders (Shen Tao,2019HB005), Yunnan Young Top Talents of Ten Thousands Plan (Shen Tao, Zhu Yan, Yunren Social Development No. 2018 73), for their support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
2. Ali, M.S.; Vecchio, M.; Pincheira, M.; Dolui, K.; Antonelli, F.; Rehmani, M.H. Applications of Blockchains in the Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1676–1717. [CrossRef]
3. Kumar, S.; Pundir, A.K. Integration of IoT and Blockchain Technology for Enhancing Supply Chain Performance: A Review. In Proceedings of the 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 4–7 November 2020; pp. 396–401.
4. Mazhar, N.; Salleh, R.; Zeeshan, M.; Hameed, M.M. Role of Device Identification and Manufacturer Usage Description in IoT Security: A Survey. *IEEE Access* **2021**, *9*, 41757–41786. [CrossRef]
5. Liang, X.; Kim, Y. A Survey on Security Attacks and Solutions in the IoT Network. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Online, 27–30 January 2021; pp. 853–859.
6. Khor, J.H.; Sidorov, M.; Woon, P.Y. Public Blockchains for Resource-Constrained IoT Devices—A State-of-the-Art Survey. *IEEE Internet Things J.* **2021**, *8*, 11960–11982. [CrossRef]
7. Nakamoto, S.J.C. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 10 January 2022).
8. Singh, S.; Hosen, A.S.M.S.; Yoon, B. Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network. *IEEE Access* **2021**, *9*, 13938–13959. [CrossRef]
9. Meng, T.; Zhao, Y.; Wolter, K.; Xu, C.Z. On Consortium Blockchain Consistency: A Queueing Network Model Approach. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1369–1382. [CrossRef]
10. Zhuang, P.; Zamir, T.; Liang, H. Blockchain for Cybersecurity in Smart Grid: A Comprehensive Survey. *IEEE Trans. Ind. Inform.* **2021**, *17*, 3–19. [CrossRef]
11. Asheralieva, A.; Niyato, D. Reputation-Based Coalition Formation for Secure Self-Organized and Scalable Sharding in IoT Blockchains with Mobile Edge Computing. *IEEE Internet Things J.* **2020**, *7*, 11830–11850. [CrossRef]
12. Sadawi, A.A.; Hassan, M.S.; Ndiaye, M. A Survey on the Integration of Blockchain With IoT to Enhance Performance and Eliminate Challenges. *IEEE Access* **2021**, *9*, 54478–54497. [CrossRef]
13. Riccia, L.; Maesab, D.; Favenzac, A.; Ferro, E.J.I.A. Blockchains for COVID-19 contact tracing and vaccine support: A systematic review. *IEEE Access* **2021**, *9*, 37936–37950. [CrossRef]

14. Yun, J.; Goh, Y.; Chung, J.M. DQN-Based Optimization Framework for Secure Sharded Blockchain Systems. *IEEE Internet Things J.* **2021**, *8*, 708–722. [CrossRef]
15. IBFT. Available online: https://github.com/jpmorganchase/quorum (accessed on 10 January 2022).
16. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. *ACM SIGMETRICS Perform. Eval. Rev.* **2014**, *42*, 34–37. [CrossRef]
17. Poet 1.0 Specification. Available online: https://sawtooth.hyperledger.org/docs/core/releases/1.0/architecture/poet.html (accessed on 26 September 2018).
18. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance. *OSDI* **2002**, *99*, 173–186.
19. Tama, B.A.; Kweka, B.J.; Park, Y.; Rhee, K. A critical review of blockchain and its current applications. In Proceedings of the 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang, Indonesia, 22–23 August 2017; pp. 109–113.
20. Choi, B.G.; Jeong, E.; Kim, S.W. Multiple Security Certification System between Blockchain Based Terminal and Internet of Things Device: Implication for Open Innovation. *J. Open Innov. Technol. Mark. Complex.* **2019**, *5*, 87. [CrossRef]
21. Ali, O.; Jaradat, A.; Kulakli, A.; Abuhalimeh, A. A Comparative Study: Blockchain Technology Utilization Benefits, Challenges and Functionalities. *IEEE Access* **2021**, *9*, 12730–12749. [CrossRef]
22. Chakraborty, S.; Aich, S.; Kim, H. A Secure Healthcare System Design Framework using Blockchain Technology. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Korea, 17–20 February 2019; pp. 260–264.
23. Guo, J.; Ding, X.; Wu, W. A Blockchain-Enabled Ecosystem for Distributed Electricity Trading in Smart City. *IEEE Internet Things J.* **2021**, *8*, 2040–2050. [CrossRef]
24. Yu, K.; Tan, L.; Aloqaily, M.; Yang, H.; Jararweh, Y. Blockchain-Enhanced Data Sharing With Traceable and Direct Revocation in IIoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7669–7678. [CrossRef]
25. Miller, D. Blockchain and the Internet of Things in the Industrial Sector. *IT Prof.* **2018**, *20*, 15–18. [CrossRef]
26. Li, G.; Fang, C.-C. Promoting Information-Resource Sharing within the Enterprise: A Perspective of Blockchain Consensus Perception. *J. Open Innov. Technol. Mark. Complex.* **2021**, *7*, 177. [CrossRef]
27. Salah, K.; Rehman, M.H.; Nizamuddin, N.; Al-Fuqaha, A.J.I.A. Blockchain for AI: Review and Open Research Challenges. *IEEE Access* **2018**, *7*, 10127–10149. [CrossRef]
28. Bach, L.M.; Mihaljevic, B.; Zagar, M. Comparative analysis of blockchain consensus algorithms. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 1545–1550.
29. Ongaro, D.; Ousterhout, J. In search of an understandable consensus algorithm. In Proceedings of the 2014 {USENIX} Annual Technical Conference, Philadelphia, PA, USA, 19–20 June 2014.
30. Lamport, L. *The Part-Time Parliament*; Concurrency: Chicago, IL, USA, 2019.
31. Jun, M. Blockchain government-A next form of infrastructure for the twenty-first century. *J. Open Innov. Technol. Mark. Complex.* **2018**, *4*, 7. [CrossRef]
32. Setyowati, M.S.; Utami, N.D.; Saragih, A.H.; Hendrawan, A. Blockchain Technology Application for Value-Added Tax Systems. *J. Open Innov. Technol. Mark. Complex.* **2020**, *6*, 156. [CrossRef]
33. Bai, F.; Shen, T.; Yu, Z.; Zeng, K.; Gong, B. Trustworthy Blockchain-Empowered Collaborative Edge Computing-as-a-Service Scheduling and Data Sharing in the IIoE. *IEEE Internet Things J.* **2021**, *8*, 6437–6453. [CrossRef]
34. Luu, L.; Narayanan, V.; Zheng, C.; Baweja, K.; Saxena, P. A Secure Sharding Protocol For Open Blockchains. In Proceedings of the 2016 ACM SIGSAC Conference, Vienna, Austria, 24–28 October 2018.
35. Zamani, M.; Movahedi, M.; Raykova, M. RapidChain: Scaling Blockchain via Full Sharding. In Proceedings of the 2018 ACM SIGSAC Conference, Toronto, ON, Canada, 15–19 October 2018.
36. Yu, G.; Wu, B.; Niu, X. Improved Blockchain Consensus Mechanism Based on PBFT Algorithm. In Proceedings of the 2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), Suzhou, China, 20–22 March 2020; pp. 14–21.
37. Li, W.; Feng, C.; Zhang, L.; Xu, H.; Cao, B.; Imran, M.A. A Scalable Multi-Layer PBFT Consensus for Blockchain. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1146–1160. [CrossRef]
38. Hu, M.; Shen, T.; Men, J.; Yu, Z.; Liu, Y. CRSM: An Effective Blockchain Consensus Resource Slicing Model for Real-Time Distributed Energy Trading. *IEEE Access* **2020**, *8*, 206876–206887. [CrossRef]
39. Khatoon, A.J.E. A Blockchain-Based Smart Contract System for Healthcare Management. *Electronics* **2020**, *9*, 94. [CrossRef]
40. Micali, S.; Rabin, M.; Vadhan, S. Verifiable random functions. In Proceedings of Symposium on Foundations of Computer Science. In Proceedings of the 40th Annual Symposium on Foundations of Computer Science, New York, NY, USA, 17–19 October 1999.
41. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying Density-Based Local Outliers. In Proceedings of the Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000.
42. Hao, X.; Yu, L.; Zhiqiang, L.; Zhen, L.; Dawu, G. Dynamic Practical Byzantine Fault Tolerance. In Proceedings of the 2018 IEEE Conference on Communications and Network Security, Beijing, China, 30 May–1 June 2018; pp. 1–8.
43. Jiang, Y.; Lian, Z. High Performance and Scalable Byzantine Fault Tolerance. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019.

44. Sakho, S.; Zhang, J.; Essaf, F.; Badiss, K.; Kiprop, J.K. Research on an improved practical byzantine fault tolerance algorithm. In Proceedings of the 2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), Suzhou, China, 20–22 March 2020.
45. Wu, Y.; Song, P.; Wang, F. Hybrid Consensus Algorithm Optimization: A Mathematical Method Based on POS and PBFT and Its Application in Blockchain. *Math. Probl. Eng.* **2020**, *2020*, 7270624. [CrossRef]
46. Li, Y.; Qiao, L.; Lv, Z. Consortium Blockchain Consensus Algorithm Based on PBFT. *Peer-to-Peer Netw. Appl.* **2021**, *48*, 133–141. [CrossRef]