HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University - Computer and Information Sciences

journal homepage: www.sciencedirect.com

Full length article

CE-PBFT: A high availability consensus algorithm for large-scale consortium blockchain

Jing Xiao [a], Tao Luo [a], Chaoqun Li [b], Jie Zhou [a], Zhigang Li [a,*]

[a] College of Information Science and Technology, Shihezi University, Shihezi 832000, China
[b] College of Computer Science and Technology, Shandong University, Qingdao, China

ARTICLE INFO

Keywords:
Blockchain
Consensus algorithm
Consortium blockchain
PBFT

ABSTRACT

The consortium blockchain has been widely applied in various fields such as agricultural product traceability, supply chain management, and logistics transportation. As an indispensable component of a consortium blockchain, the consensus algorithm ensures the consistency and trustworthiness of each node in the network. However, existing consensus algorithms in large-scale consortium blockchain scenarios suffer from low system throughput and high latency due to the complexity of communication processes, rendering them impractical for real-world use. To address these issues, this paper proposes a novel consensus algorithm called credit evaluation-based practical Byzantine fault tolerance (CE-PBFT). This algorithm designs a new node credit evaluation model that considers node completion rate, consensus decay, and node behavior. It effectively measures and reflects the specific reliability status of nodes during system operation, thereby enhancing system reliability and security. Additionally, the paper introduces the innovative use of decision tree algorithms to analyze network node behavior and simplifies the existing consensus protocol. Nodes are categorized as excellent, good, ordinary, or poor based on the classification results, and non-Byzantine nodes are dynamically selected accordingly. This greatly improves the overall efficiency of the system. The performance of CE-PBFT is validated through experiments and compared with PBFT, G-PBFT, RBFT, WBFT and PPoR. Experimental results demonstrate that in large-scale consortium scenarios, CE-PBFT significantly improves system throughput, effectively reduces transaction latency and communication overhead, and outperforms the compared protocols.

1. Introduction

Blockchain is classified as a form of distributed ledger technology, (Zhao et al., 2021), originally proposed by the creator of bitcoin to support the decentralized digital currency system of bitcoin. Over time, blockchain technology has gradually developed and been widely used, not only in the field of cryptocurrency. The development of blockchain has gone through several stages (Yang, 2019; Nuttah et al., 2023). At first, blockchain is mainly used to support the transaction and accounting functions of cryptocurrency. As the first blockchain application, bitcoin has led the rise of cryptocurrency and the exploration of blockchain technology. Subsequently, people began to realize the potential application value of blockchain technology and began to explore applications in other fields. Nowadays, blockchain has been applied in many fields (Rahman et al., 2021; Jindal et al., 2020; Song et al., 2021), including finance (Saba et al., 2023), supply chain management (Vangala et al., 2021) and health care (Myrzashova et al., 2023).

According to the permissions and accessibility of participants, blockchain can be divided into public blockchain (Tian et al., 2019; Khor et al., 2023), consortium blockchain (Bai et al., 2022) and private blockchain (Wang et al., 2023; Chen et al., 2022b). The consortium blockchain, composed of a group of known and trusted nodes, establishes cooperative relationships among one or more entities, organizations, or enterprises. Its role is to provide participants with a secure,

* Corresponding author.
E-mail addresses: xj_inf@shzu.edu.cn (J. Xiao), 1150063214@qq.com (T. Luo), 894068271@qq.com (C. Li), jiezhou@shzu.edu.cn (J. Zhou), Lzg_inf@shzu.edu.cn (Z. Li).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2024.101957
Received 30 October 2023; Received in revised form 27 January 2024; Accepted 28 January 2024
Available online 7 February 2024

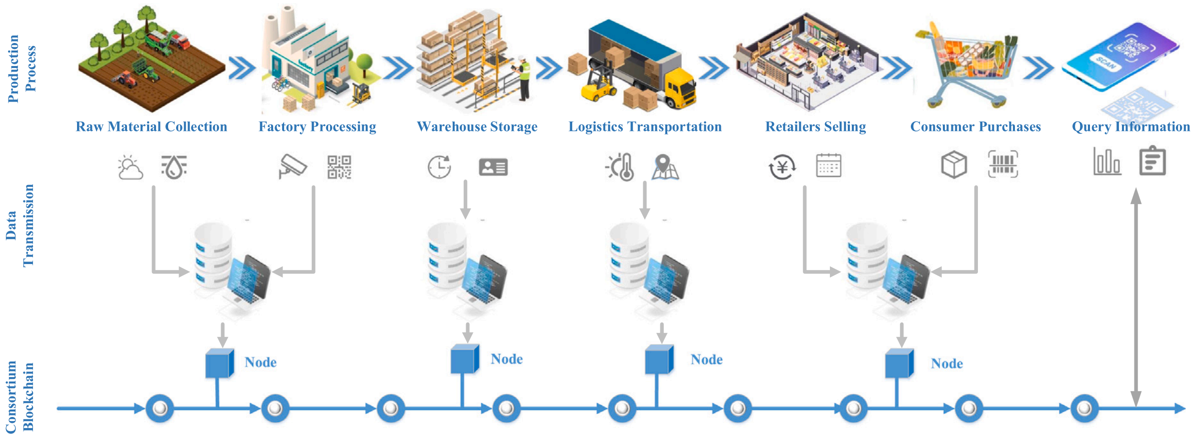

Fig. 1. Schematic diagram of blockchain used in traceability field.

efficient, and trusted environment, promoting the development of information sharing and transactions (Boateng et al., 2022). Consortium blockchains have characteristics such as multi-party participation, restricted access permissions, and high privacy protection. Additionally, they offer advantages in terms of traceability and auditability, enabling easy tracing of transaction origins and history. By selecting appropriate consensus mechanisms, consortium blockchains can provide higher performance and scalability for processing transactions and data, meeting the specific needs of organizations or industries.

Because it is suitable for cooperation and data sharing among multiple organizations, consortium blockchain has shown good application prospects in many fields. At present, it is widely used in the fields of agricultural traceability (Misra et al., 2022), supply chain management (Viriyasitavat et al., 2022), Internet of Things communication (Li and Shi, 2023), logistics and transportation (Li et al., 2020). Taking the agricultural traceability field as an example, consortium blockchain can provide real-time data sharing and collaborative mechanisms, reducing information asymmetry and cumbersome procedures in traditional agricultural traceability. This, in turn, enhances the efficiency of agricultural product traceability, while reducing labor and time costs (Guan et al., 2023). At the same time, consortium blockchain technology records the whole process of agricultural products from production to consumption, including planting, processing, transportation and other links. By sharing data and smart contracts to ensure that participants jointly verify and record transaction information and provide reliable data, we can realize the whole process traceability of agricultural products and improve the traceability, transparency and credibility of products. Some agricultural products enterprises and organizations have established traceability platforms using consortium blockchain technology to record the production, processing, transportation and other information of agricultural products, and ensure the credibility of data through blockchain technology. Through the traceability system of the consortium blockchain, consumers can obtain real information and proof of agricultural products, and enhance their sense of trust in products. The traceability information of agricultural products can be displayed to consumers by scanning QR codes and viewing transaction records on the blockchain, so as to enhance the brand value and market competitiveness of products. Specially, schematic diagram of blockchain used in traceability field is given in Fig. 1.

As a crucial component of consortium blockchain networks, the consensus algorithm plays a vital role in ensuring the consistency of transactions and states in the blockchain's core features by coordinating the various nodes in the network (Kang et al., 2019). However, current consensus algorithms commonly face a series of limitations and challenges.

Consensus algorithms may encounter performance bottlenecks when handling large-scale networks. With an increasing number of nodes and expanding network scale, consensus algorithms require more computational and communication resources to achieve consensus, resulting in performance degradation (Zhang et al., 2022). Issues related to processing latency and throughput become more prominent, affecting the speed and efficiency of transaction processing.

Furthermore, the security of consensus algorithms is also a concern (Sun et al., 2021). The presence of malicious nodes can pose threats to the consensus process by engaging in disruptive behavior that compromises consensus consistency or manipulates transaction data. Therefore, consensus algorithms need to possess mechanisms to withstand various attacks and protect network security. However, some consensus algorithms may have insufficient security in certain scenarios, leading to vulnerabilities such as data tampering, thereby undermining the overall system's trustworthiness and security.

In the face of these limitations and challenges, researchers are actively seeking solutions. In recent years, consensus algorithms based on reputation value have gradually become a research focus. These algorithms assess the reputation value of nodes to determine their authority and trustworthiness in the consensus process. By establishing a trust mechanism and mutual trust relationships between nodes, they effectively prevent double-spending attack, Byzantine attack, 51% attack, and Sybil attacks, ensuring the correctness, consistency, and security of consensus. Through reputation-based consensus algorithms, consortium chains can enhance the reliability of transactions, prevent data tampering and malicious behavior, and strengthen the overall system's security and trustworthiness. Additionally, they have the potential to improve the performance and scalability of consortium blockchain networks, enabling them to adapt to the growing network scale.

In the context described, this paper proposes a novel consensus algorithm, CE-PBFT, by considering factors such as node completion rate, consensus participation level, and node behavior. The algorithm introduces a novel node credit evaluation system that effectively assesses the reliability of nodes during system operation. This approach not only enhances system reliability but also strengthens its security. Additionally, leveraging the ID3 decision tree algorithm innovatively, which possesses characteristics such as strong interpretability, adaptability to discrete features, handling missing values, and high efficiency, the proposed algorithm analyzes network node behavior and simplifies existing consensus protocols. Through this approach, nodes are categorized as excellent, good, ordinary, or poor, and non-Byzantine nodes are dynamically selected based on these classifications, significantly improving the overall operational efficiency of the system.

Specifically, the main contributions of this paper are as follows:

(1) In this paper, a novel consensus algorithm for consortium blockchains termed Credit Evaluation-based Practical Byzantine Fault Tolerance (CE-PBFT) is proposed. This algorithm incorporates a new node credit assessment model and leverages the ID3 decision tree

algorithm to analyze the behaviors of consortium blockchain nodes, effectively addressing the challenges in large-scale consortium blockchain scenarios.

(2) A novel node credit evaluation system has been designed and implemented, taking into account metrics such as node completion rates, consensus decay, and node behavior indices. This system accurately measures the credibility and reliability of nodes within the operational blockchain system. Innovatively, the ID3 decision tree algorithm is employed to categorize nodes into four classes: excellent, good, average, and poor. Based on these classifications, non-Byzantine nodes are dynamically selected, significantly enhancing the system's security.

(3) Furthermore, the consensus protocol has been simplified, optimizing the communication overhead within the protocol and substantially improving the overall performance of the system. Additionally, the view change protocol has been refined to restrict the selection of Primary Nodes (PN) to those with a high level of node credibility, thus effectively ensuring the system's reliability.

(4) The performance of CE-PBFT is empirically validated through a series of experiments presented in this paper. The experimental results demonstrate that, compared to PBFT, G-PBFT, RBFT, WBFT and PPoR, CE-PBFT offers significant advantages in terms of system throughput, transaction latency, and communication overhead in large-scale consortium scenarios.

The remaining sections of this article are organized as follows. In Section 2, related work is described, including a brief overview of blockchain technology and some of the most popular consensus algorithms. Section 3 introduces the original PBFT consensus algorithm. The proposed consensus algorithm will be described in Section 4. Experimental results and related discussions can be found in Section 5. The conclusion and future research directions will be provided in Section 6.

2. Related work

As an indispensable part of the consortium blockchain network, the consensus algorithm is the cornerstone to ensure the safe, reliable and consistent operation of the blockchain networks (Xu et al., 2019). Through consensus algorithms, blockchain network can prevent a variety of malicious behaviors and attacks. Under the mechanism of consensus algorithm, the participants in the network must reach an agreement and verify the effectiveness of the transaction before they can be added to the blockchain. In this way, any participant who attempts to carry out malicious operations will be rejected by other honest participants. At the same time, all participants can see the same facts and states without differences or conflicts. This not only ensures the reliability of the blockchain, but also ensures the consistency of the blockchain network. Many researches on blockchain have proposed a variety of consensus algorithms. This section will review the existing consensus algorithms.

Proof of X (PoX) is a classical consensus protocol, in which X represents different attributes, proof of work (PoW), proof of stake (PoS), and proof of space. These protocols are different from the traditional centralized consensus mechanism, which ensures consistency of the system by providing specific proof by participants. PoW is the first consensus algorithm applied to blockchain (Feng and Luo, 2020). In PoW, participants need to solve complex mathematical problems to obtain authoritative block verification rights, but this problem requires a lot of computing power and energy consumption. PoS determines its authority in the consensus process based on the number of cryptocurrencies held by participants (Cao et al., 2020). Unlike PoW, PoS consumes less energy and is more environmentally friendly. Delegated proof of stack (DPoS) is an improved consensus algorithm (Liu et al., 2021). Its core idea is to verify and package transactions by electing representatives, so as to determine the consensus state of the blockchain. DPoS aims to solve the problems of resource centralization and performance scalability in traditional PoS algorithm. PoX (proof of X) consensus algorithm has been widely used in the field of blockchain, but they also have some shortcomings and deficiencies. PoX algorithm may lead to the situation that a few large participants control the network. In this case, resource centralization may weaken the goal of decentralization and increase potential security risks. In addition, PoX has the problems of low efficiency, high computing power and high energy consumption. This makes PoX algorithm unable to adapt to various actual use scenarios of consortium blockchain.

Byzantine fault tolerance (BFT) and Practical Byzantine fault tolerance (PBFT) are two common solutions based on consensus algorithm, aiming to solve the Byzantine fault tolerance problem in distributed systems (Ge et al., 2022; Misic et al., 2021). They also play a crucial role in ensuring the security and consistency of blockchain networks. BFT is a paradigm of consensus algorithm, which is used to enable the distributed system to operate normally in the presence of Byzantine errors (such as node failure, malicious behavior, etc.). PBFT is a specific BFT algorithm, which provides a practical solution. The core idea of PBFT algorithm is to reach a consensus through the interaction of multiple replica nodes, and requires a few nodes to correctly reach a consensus decision. PBFT algorithm can tolerate a certain number of Byzantine error nodes, so it has certain fault tolerance. However, PBFT algorithm needs complex message interaction for each request, which makes it more expensive and requires higher transaction latency and bandwidth.

Due to these problems in the practical application of PBFT consensus algorithm, many scholars have improved all aspects of PBFT (Misic et al., 2023). To reduce the latency of the consensus process of the Internet of vehicles and optimize the transaction frequency of the consensus process, Kumar et al. (2023) designed a fast and intelligent repute PBFT consensus protocol R-PBFT, which uses the reputation obtained by logical regression to optimize the process of PBFT consensus aggregation, and finally can use this protocol to reduce the load pressure of the vehicle network server. Aiming at the problem of low node scalability of PBFT in large-scale networks, Li et al. (2021) designed the best two-layer PBFT by means of node hierarchical communication, and finally constructed a scalable multi-layer PBFT consensus protocol, reducing the complexity of protocol communication. Inspired by PoS, Qushtom et al. (2023) uses the principle of PoS to optimize the PBFT consensus mechanism, and obtains a two-stage PBFT consensus protocol based on trust score and reward mechanism. Using trust score and reward mechanism, we can effectively motivate honest nodes and effectively deal with dishonest nodes. Chen et al. (2022a) based on the improvement of PBFT, feature grouping and credit optimization Byzantine fault tolerance (FCBFT) is proposed. Through the designed feature grouping model, the nodes of the large-scale consortium blockchain are divided into different institutions, and the reputation score reward mechanism is further designed. The main node is selected by using the reputation, which enhances the efficiency of the consensus aggregation process. Using the proposed score grouping mechanism to improve the traditional PBFT, Xu et al. (2022) designed a new score grouping PBFT protocol, which shortened the time of the consensus aggregation process and improved the vehicle authentication efficiency and robustness of the vehicle network. According to the reputation of platform users evaluated by the designed chain code, Tong et al. (2022) proposed a new consensus protocol, reputation based PBFT, and further applied it to the designed hybrid blockchain crowdsourcing platform, which improved the transaction throughput of the platform and optimized the fault tolerance of the platform. However, the above consensus algorithm not only pursues consistency and security, but also sacrifices the performance of the system. Due to the complex messaging and consensus protocols, these algorithms may show low throughput and latency when dealing with numerous transactions or numerous nodes. Simultaneously, with the growth of the scale of distributed systems, these algorithms are facing the challenge of scalability. It may not be able to scale effectively in large-scale networks, resulting in

Table 1
Description of characters.

Character	Description
$CE - PBFT$	Credit evaluation based Practical Byzantine Fault Tolerance algorithm
ID3 decision tree	Iterative Dichotomiser 3 decision tree
PN	Primary node
$Comp$	Node completion rate
$comp\_tms$	The number of consensus a node completes
$total\_tms$	The total number of consensus
$\psi(z)$	Consensus decay
ζ	Impact factor
$Credit_i$	Credit evaluation value of node i
$Value_i(total\_tms)$	Score obtained by node i during the $total\_tms$ round of consensus
S	The set of all sample
T	Feature vector
S^v	The v_{th} value of attribute T in S
m	The number of attributes
n	The number of categories
$\lvert S^v \rvert$, $\lvert S \rvert$	The number of samples in S^v and S sets
$IE(S)$	Information entropy
$IG(S, T)$	Information gain
F	The maximum number of nodes allowed to fail in all nodes

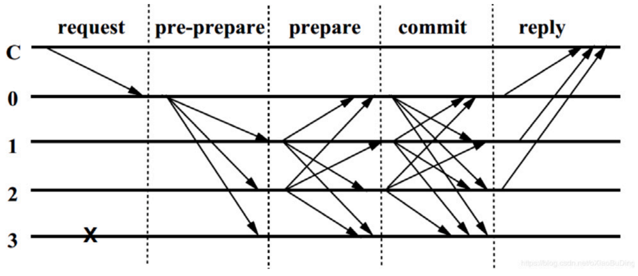

Fig. 2. Consistency protocol of PBFT.

performance degradation or unable to cope with the nodes growing rapidly.

To address the aforementioned issues, a novel improved Byzantine consensus algorithm is designed based on decision trees and credit evaluating mechanism. In Section 4, the algorithm will be described detailed.

3. Overview of PBFT

The blockchain consensus algorithm PBFT is a consensus algorithm based on Byzantine fault tolerance, which aims to solve the Byzantine fault problem in distributed systems. It achieves consensus through mutual cooperation between nodes, and ensures that the system can maintain security and consistency in the face of attacks by malicious nodes. The principle of PBFT is based on the Byzantine general problem, which is a classic distributed system problem. A group of members need to reach a consensus on a certain action, but some members may be malicious or untrusted. PBFT solves this problem by introducing the cooperation and verification of multiple nodes.

The core idea of PBFT is to reach consensus through mutual verification and message exchange between nodes. The node will verify the messages from the PN and other nodes, and judge the legitimacy and consistency of the messages according to certain rules. Only when a node receives a sufficient number of legitimate messages will it enter the next stage and broadcast its own messages to other nodes. Through the cooperation and verification of multiple nodes, PBFT can reach a consensus and ensure that the system can still maintain security and consistency in the face of Byzantine faults. Specifically, it ensures system consistency and fault tolerance through consistency protocol, view change protocol and checkpoint protocol.

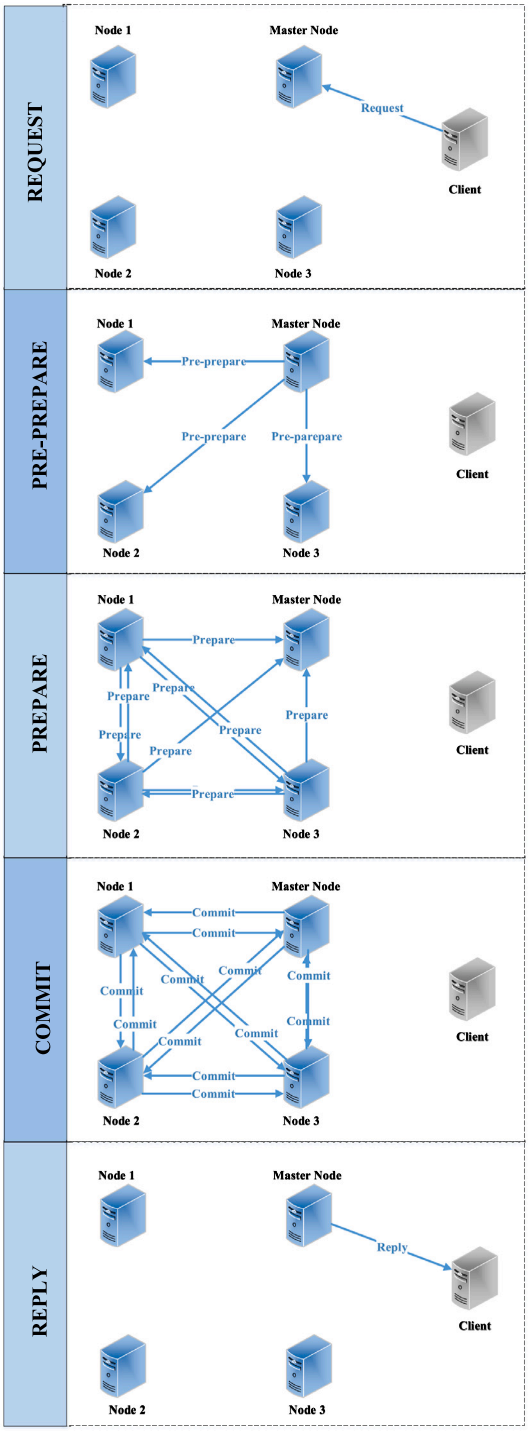

Fig. 3. Schematic diagram of PBFT algorithm operation.

3.1. Consistency protocol

The consistency protocol of PBFT protocol is its core component, which is used to ensure that all replica nodes in the system agree on the order and result of requests. The consistency agreement of PBFT is divided into four stages: pre-prepare, prepare, commit and reply. To better illustrate the protocol, the consistency protocol diagram of the algorithm is shown in Fig. 2 and schematic diagram of PBFT algorithm operation is given further in Fig. 3.

In the pre-prepare phase, the client sends the request to the PN, and the PN broadcasts the request to other replica nodes. The pre-prepare message contains information such as the summary and serial number of the request. Other replica nodes verify the validity of the request and record the pre-prepare message. Next is the prepare phase. After receiving the pre-prepare message, the replica node will broadcast the prepare message to other replica nodes, including the confirmation of the pre-prepare message. The prepare message contains the summary of the request, the serial number, and the commitment to the message. When the replica node receives a sufficient number of prepare messages, it can enter the next stage. The commit phase is triggered after the replica node receives a sufficient number of prepare messages. The replica node sends a commit message to other nodes, indicating that the request is confirmed. The commit message contains the summary of the request, the serial number, and the confirmation of the message. When the replica node receives a sufficient number of commit messages, it executes the result of the request and sends a reply message to the client. The last is the reply phase. After executing the request, the replica node sends a reply message to the client, including the result of the request. After receiving a sufficient number of reply messages, the client can confirm that the request has been processed and continue to send the next request.

Through this series of phases and message interaction, PBFT protocol ensures that all replica nodes agree on the order and results of requests. Only after a sufficient number of replica nodes confirm and accept the request, the system will proceed to the next step. This mechanism enables the PBFT protocol to tolerate Byzantine errors, and maintain consistency and correctness even if some replica nodes fail or behave maliciously. It is suitable for scenarios that need to ensure consistency and reliability in distributed systems.

3.2. View change protocol

The view change protocol of PBFT protocol is designed to solve the failure or crash of the PN. When the replica node detects the failure of the PN, it needs to elect a new PN through the view change protocol to ensure the normal operation and consistency of the system. The process of view change agreement includes view proposal, proposal and confirmation, and view change.

First, when the replica node detects the failure of the PN, it will broadcast a view proposal message containing the proposed new view number. These messages are propagated to other replica nodes, triggering the election process of the new view. Next, after receiving the view proposal message, the replica node will confirm the proposal under certain conditions. These conditions may include receiving a sufficient number of consent messages, or waiting for a period of time to ensure that a sufficient number of replica nodes have received the proposal. When a sufficient number of replica nodes confirm the same view proposal, they can enter the next view. Once the new view is confirmed, the replica node will update the view number and re-enter the pre-prepare phase. The new PN will be responsible for coordinating the messaging and consensus processes of the replica node. In this way, the system can continue to operate normally and maintain consistency in the event of a PN failure.

The key of view change protocol is to ensure that a sufficient number of replica nodes can reach an agreement on the new view. Through voting and message exchange, the replica node can reach a consensus and select a new PN. In this way, PBFT protocol can realize fast and reliable view change in the face of PN failure, and ensure the continuity and consistency of the system. As a mechanism designed to solve the failure of the PN, the view change protocol elects a new PN through the message exchange and consensus mechanism between the replica nodes, and ensures that the system can continue to operate normally and maintain consistency in the case of the failure of the PN. This makes PBFT protocol have high fault tolerance and reliability.

3.3. Checkpoint protocol

The checkpoint protocol of PBFT protocol is a mechanism introduced to reduce the overhead of state transmission. It periodically saves the checkpoint status of the system and ensures consistency through mutual verification. The process of checkpoint protocol includes checkpoint proposal, checkpoint message and checkpoint verification. First, within a certain time interval, the PN sends a checkpoint proposal message to the replica node, requiring the replica node to save the checkpoint status of the current system. The checkpoint proposal message contains the checkpoint serial number and related information.

After receiving the checkpoint proposal message, the replica node will verify the validity of the proposal and save the checkpoint status if the conditions are met. The conditions for verification may include ensuring that the serial number proposed by the checkpoint is in ascending order, and that the replica node has saved all the request results before the checkpoint. Once the replica node has saved the checkpoint status, it will send a checkpoint message to other replica nodes, including the checkpoint serial number and verification information. After receiving the checkpoint message, other replica nodes will verify the checkpoint and send the checkpoint verification message to the replica node that has passed the verification. When the replica node receives a sufficient number of checkpoint verification messages, it marks the checkpoint status as completed. In this way, in the process of fault recovery, the system can directly start from the latest checkpoint state without transmitting the execution results of all historical requests.

Based on checkpoint protocol, PBFT protocol can reduce the amount of state transmission in the process of fault recovery, and improve the efficiency and performance of the system. The replica node only needs to transmit the request results from the latest checkpoint state to the current state, without transmitting all the historical request results. This reduces the use of network bandwidth and transmission delay, so that the system can recover state and achieve consistency faster.

PBFT ensures the consistency of the replica node to the request through the consistency protocol, solves the problem of the PN failure through the view change protocol, and reduces the state transmission overhead through the checkpoint protocol. Based on these protocols, PBFT can achieve the consistency of component cloth networks under the condition of tolerating Byzantine errors. However, PBFT has certain restrictions on the number of nodes. In order to ensure the security of the system, at least two-thirds of the nodes need to be honest. This means that when the number of nodes is large, it may become difficult to ensure that more than two-thirds of the nodes are reliable, which may lead to the performance degradation of PBFT. In addition, PBFT has high requirements for communication between nodes and requires a certain network bandwidth and delay. When the number of nodes is large or the network conditions are poor, the performance and security of PBFT will be greatly affected.

4. The proposed CE-PBFT algorithm

This research aims to design a consensus algorithm with low communication overhead and high availability. Therefore, an improved Byzantine fault-tolerant algorithm using Iterative Dichotomiser 3 (ID3) decision tree classification is proposed, which is called CE-PBFT. The algorithm simplifies the consistency protocol and optimizes the communication overhead. Meanwhile, the reputation integral mechanism is introduced. After the consensus process, the mechanism counts the number of continuous consensus, downtime, error communication and node activity level. and uses the decision tree classification algorithm to divide the nodes into high-quality nodes, good nodes, ordinary nodes and low-quality nodes according to these node attributes. Specially, node activity level is obtained by the ratio of node response times to the number of received messages. This can dynamically adjust the node identity and eliminate malicious nodes in the system, so as to improve the reliability of the node. In addition, the view change protocol is improved and high-quality nodes with high reputation are selected in the system as PN and candidate node. According to the function of the node, the node is divided into PN, candidate node and consensus node. In particular, Table 1 lists the characters in this paper.

PN is the node responsible for producing blocks and processing client instructions. It replies to the client to indicate that the consensus is complete. In CE-PBFT algorithm, the selection of PN is based on the node with the highest reputation score among excellent nodes. When PN functioning without any issues, the candidate node is responsible for receiving and responding to the messages sent by the PN and participating in the consensus process. When the PN has problems, the candidate node becomes a new PN to start the consensus process. Generally, the remaining excellent nodes based on the selection of PN are regarded as candidate nodes. The consensus node is responsible for receiving and responding to PN messages and participating in the consensus process. Consensus nodes are composed of good nodes and candidate nodes.

4.1. Node reputation calculation

The calculation of node reputation points needs to be judged by node behavior, which will be affected by various factors. Node behavior can be reflected by the completion of node consensus, the performance of node consensus behavior and other factors. The core of the improved algorithm is to calculate the node reputation score by considering the influencing factors, and score and divide the nodes. Next, the influencing factors of reputation points will be described in detail.

Node completion rate refers to the proportion of nodes participating in the consensus process and successfully completing the consensus, expressed in $Comp$. This reflects the situation in which nodes participate in consensus in the system.

$$Comp = \frac{comp\_tms}{total\_tms + 1} \tag{1}$$

In formula (1), $comp\_tms$ represents the number of consensus a node completes, and $total\_tms$ represents the total number of consensus. The value of $Comp$ increases with the ratio of $comp\_tms$ to $total\_tms$. The larger the value of $Comp$, the better the performance of the node.

Consensus decay measures the impact of consensus time on node reputation score. It reflects the impact of the time of participating in consensus on reputation decay. Function for consensus attenuation $\psi(z)$ The specific calculation method refers to formula (2).

$$\psi(z) = e^{\frac{-z}{\zeta}} \tag{2}$$

where z is the interval between the number of times of this consensus and the previous consensus. The parameter ζ is the impact factor, which is used to adjust the impact change degree. The value of ζ is $1/ln2$, so that the consensus attenuation degree is greatly affected by. Obviously, the value of ψ decreases with the increase of z. The smaller the value of ψ, the farther the previous consensus of the node is from this consensus, and the higher the attenuation degree.

$$\begin{aligned} Credit_i &= \psi(z) \cdot Value_i(total\_tms + 1) \\ &= \begin{cases} \psi(z) \cdot (Value(total\_tms) + S_{reward} \cdot Comp) & \text{node } i \text{ completes consensus} \\ \psi(z) \cdot (Value(total\_tms) + S_{punish} \cdot (1 - Comp)) & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

Node behavior performance refers to the performance of nodes. Node behavior is divided into two categories: normal participation consensus and non-participation consensus. For nodes that normally participate in consensus, the reward reputation scores rew will be given. For nodes that are absent from the consensus process, a penalty reputation score of pen will be given. These scores will affect the calculation of reputation points of nodes. The reputation score of a node is affected by various factors of node behavior. The node is comprehensively evaluated. Formula (3) given the evaluation of reputation score.

In formula (3), $Credit_i$ denotes the credit evaluation value of node i, and $Value_i(total\_tms)$ denotes the score obtained by node i in the $total\_tms_{th}$ consensus. When node i completes the consensus in the $total\_tms_{th}$ consensus, the calculation of reputation score is a comprehensive evaluation method to pursue a more accurate node situation. The algorithm design of node reputation integral calculation is as follows.

4.2. Node classification

The interpretability, adaptability, automatic feature selection, robustness, and efficiency of the decision tree in machine learning make it suitable for the task of classifying consortium blockchain nodes. The ID3 decision tree algorithm is a machine learning and data mining algorithm primarily used for classification problems. It constructs a decision tree by recursively selecting the optimal features to partition the dataset. The core of the ID3 algorithm lies in using information gain as the criterion for feature selection. Information gain is based on the concept of entropy, which measures the uncertainty of the data. By selecting the feature with the highest information gain, it can rapidly reduce the uncertainty of the data.

Applying the ID3 decision tree algorithm to node classification in consortium blockchain consensus algorithms can provide significant interpretability to the classification model. It can intuitively display the decision rules, allowing the classification process of nodes to be explained and understood. This is particularly valuable for consortium blockchain applications in traceability scenarios. Moreover, the ID3 algorithm is well-suited for handling discrete features, which are prevalent in consortium blockchain consensus algorithms. Additionally, the algorithm can handle missing values, enabling classification even when node information is incomplete. Finally, the ID3 algorithm has low computational complexity and high execution efficiency, which is beneficial for node classification in consortium blockchain consensus algorithms.

Therefore, compared to other methods of node classification in consensus algorithms, the ID3 decision tree algorithm possesses irreplaceable advantages. Compared to the K-Nearest Neighbors (KNN) algorithm, the ID3 algorithm can handle discrete features and generate decision trees with better interpretability. Compared to support vector machines (SVM), the ID3 algorithm has lower computational complexity, while SVM requires complex parameter tuning and kernel function selection. Furthermore, compared to neural networks, the ID3 algorithm does not require large amounts of training data or iterative processes, making it easier to implement and understand.

Applying the ID3 decision tree algorithm to node classification in consortium blockchain consensus algorithms offers advantages such as strong interpretability, adaptability to discrete features, handling of missing values, and high efficiency, supporting the security and reliability of consensus algorithms. In Fig. 4, a schematic diagram of the node classification process is described. Meanwhile, the flowchart of CE-PBFT is given in Fig. 5.

In the improved CE-PBFT consensus algorithm, a decision tree classification algorithm is introduced. When the system completes each round of consensus, a series of feature attributes, including reputation points, downtime times, consecutive consensus times, incorrect communication times, and node activity level will be collected by nodes. These attributes are used as features, while excellent nodes, good nodes, ordinary nodes, and poor nodes are category attributes. Firstly, the consortium blockchain collects and classifies classification attributes. Then, the main node is elected through the view change protocol, and the nodes are verified during the process. Finally, collect node behavior and eliminate poor nodes. The removed Byzantine nodes cannot participate in the following formula process, and if they need

Table 2
Permission of different kind of nodes.

	Excellent nodes	Good node	Ordinary node	Worst node
View Change Protocol	Yes	No	No	No
Consistency Protocol	Yes	Yes	No	No
Checkpoint Protocol	Yes	Yes	Yes	No

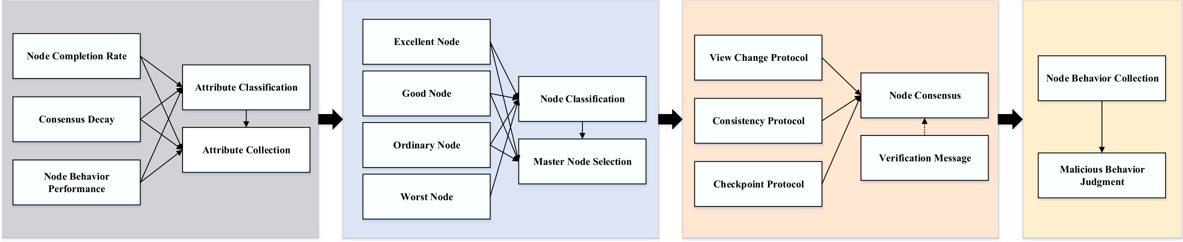

Fig. 4. Schematic diagram of node classification process.

to re-enter the system, confirmation from the consortium blockchain administrator is required.

The first step in constructing a decision tree requires calculating the information entropy, which is an indicator used to measure whether the sample set belongs to the same category. Assuming that the probabilities of the samples in the current training set belonging to the categories of excellent nodes, good nodes, ordinary nodes, and poor nodes are P1, P2, P3, and P4, respectively, the category information entropy calculated based on the training set can be expressed as formula (4).

$$IE(S) = -\sum_{i=1}^{n} p_i log_2 p_i \tag{4}$$

where n denotes the number of categories.

Secondly, information gain should be calculated, which represents the degree of uncertainty in the information. In the training set, we select the attribute feature vector T to partition the set. This feature vector T has 5 possible values, namely node reputation evaluation score, number of consecutive consensus attempts, number of downtime attempts, number of incorrect communication attempts, and node activity. According to the information entropy calculated by formula (4), the information gain calculation method for dividing the set S with the attribute feature vector T is shown in formula (5).

$$IG(S,T) = IE(S) - \sum_{v=1}^{m} \frac{|S^v|}{|S|} IE(S^v) \tag{5}$$

where S is all sample sets, S^v represents the v_{th} value of attribute T in S, and $|S^v|$ and $|S|$ respectively represent the number of samples in S^v and S sets.

Based on the ID3 decision tree generated by the above process, the nodes in the system are classified, and each node belongs to one of three categories: excellent node, good node, and ordinary node. Among them, the number of excellent nodes accounts for 20%, the number of good nodes accounts for 30%, and the number of ordinary nodes accounts for 50%. The protocols that these three types of nodes can participate in are shown in Table 2.

The level of nodes is dynamic. When adding a new system node, the algorithm will allocate an initial integral of 70 to it. The continuous consensus times, downtime times, error communication times and node activity of the new node are all 0. New nodes can improve their node level by completing consensus. After the consensus process is completed, the reputation score of all nodes in the system will be increased by 1, and the continuous consensus times, downtime times, and error communication times of each node will be counted, and the node activity will be calculated. Then wait for the next node identity adjustment. If a good node makes Byzantine errors in the process of consensus, that is, the node transmits wrong information, its reputation score will be reduced by 5, and will be reduced to ordinary nodes after the next ID3 decision tree classification. When a Byzantine error occurs at the PN, other nodes will directly remove the node from this round of consensus and reduce it to an ordinary node. The eliminated node will become the last node in the normal node. The system dynamically adjusts the identity of nodes by regularly running ID3 decision tree classification algorithm. When the excellent nodes in the system exceeds 20% in the system, these nodes will stop becoming excellent nodes and serve as the nodes with the highest serial number among the good nodes. They wait for the system to automatically upgrade to excellent nodes when the number of excellent nodes is less than 20% due to the addition or deletion of nodes. The adjustment methods of other nodes are similar.

4.3. Consistency protocol

CE-PBFT preserves all stages of PBFT and removes the communication process between network nodes, thus reducing the use of network bandwidth. The algorithm selects good nodes in the system as consensus nodes to ensure the reliability and security of nodes in the system. The consistency protocol diagram of the algorithm is shown in Fig. 6.

In the request stage, a request $\langle REQUEST, O, T, C\rangle$ will be sent to PN P by client C, which includes the requested state machine O, timestamp T and client number C.

In the pre-preparation stage, the PN P receives the proposal numbered N from the client and generates the prepare proposal $\ll \langle PREPREPARE, V, N, DIGEST\ >, OUTCOME\ > CREDIT, MESSAGE\rangle$, including the trial view number V, proposal number N, message digest $DIGEST$, hash calculation result $OUTCOME$, node credit score $CREDIT$, $MESSAGE$ and other information. It is worth noting that $OUTCOME$ refers to the hash calculation result of node reputation score credit. The purpose of recording $OUTCOME$ is to carry out subsequent verification and confirmation during the consensus process. The main node sends the prepare proposal to all consensus nodes.

After receiving the pre-preparation message sent by the PN, in the preparation stage, the consensus node confirms the correctness of the message and a preparation message $\langle PREPARE, V, N, DIGEST, I, CREDIT\rangle$ will be generated, including the type of preparation message $PREPARE$, trial view number V, proposal number N, message summary $DIGEST$, node number I and reputation score $CREDIT$. After the PN receives the preparation messages sent by other consensus nodes, if it receives more than $(2F + 1)$ messages, it will enter the confirmation phase. Here, F represents the maximum number of nodes allowed to fail. If the number of preparation messages received is insufficient, the PN thinks that an error has occurred, gives the PN

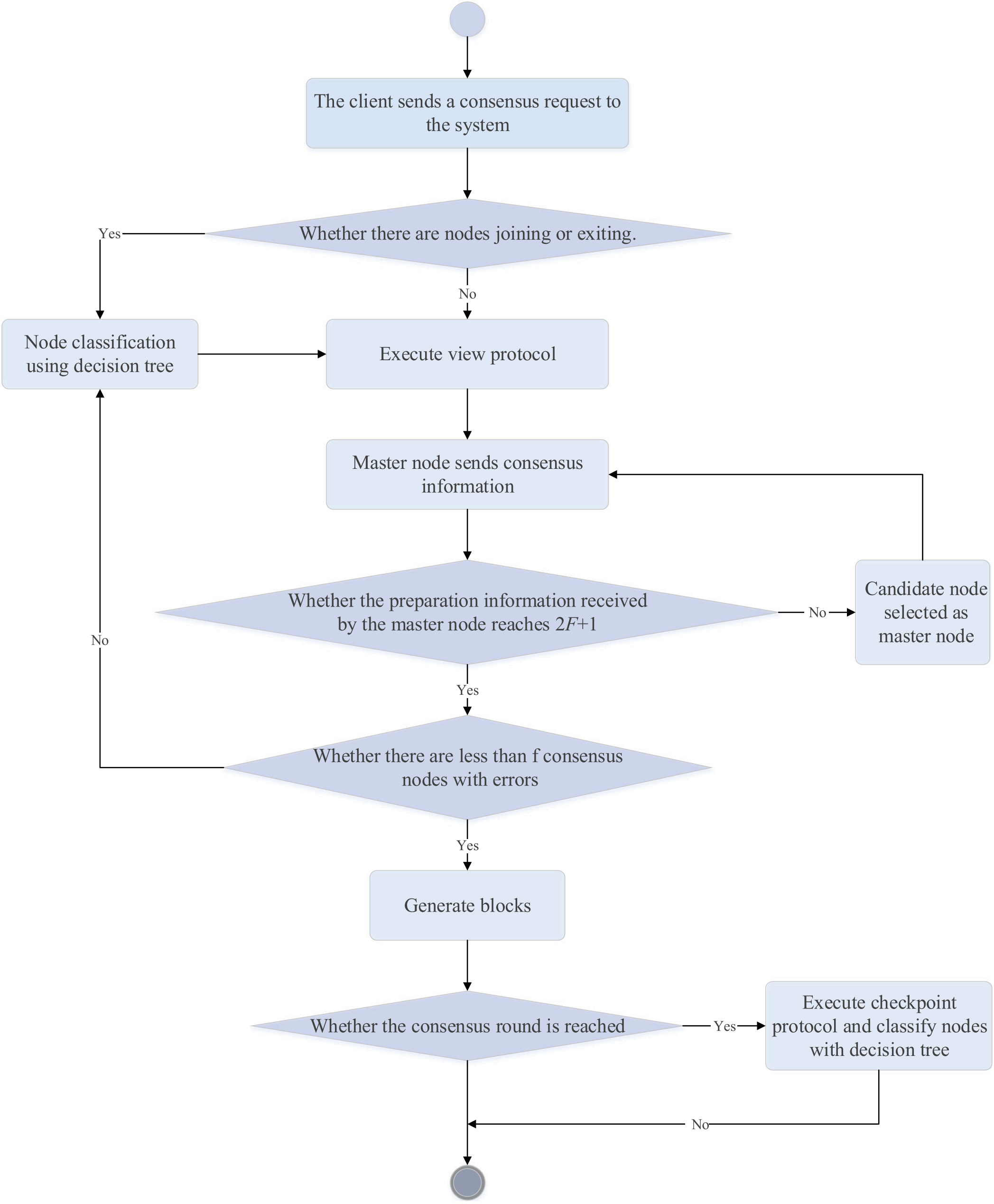
The client sends a consensus request to the system
Whether there are nodes joining or exiting.
Yes
No
Node classification using decision tree
Execute view protocol
Master node sends consensus information
Whether the preparation information received by the master node reaches 2F+1
No
Candidate node selected as master node
Yes
No
Whether there are less than f consensus nodes with errors
Yes
Generate blocks
Whether the consensus round is reached
Yes
Execute checkpoint protocol and classify nodes with decision tree
No

Fig. 5. Flowchart of CE-PBFT algorithm.

permission to the newly elected PN, and sends the message of consensus verification failure $\langle FAIL, V, T, M, K\rangle$ to all nodes, where the view number V is plus one. The PN with errors is reduced to a normal node.

Then in the confirmation stage, after receiving the preparation messages sent by the consensus node, the PN P checks whether all the messages are correct and the same, and generates confirmation information $\langle COMMIT, V, N, DI- GEST, CONFIRM\rangle$, including view number V, proposal number N, message digest $DIGEST$ and confirmation flag $CONFIRM$. The main node broadcasts the confirmation information to all consensus nodes and informs the client that the new block has been confirmed to be added. If there are less than F consensus nodes sending messages that are inconsistent with the messages of the PN, continue to send confirmation messages and generate new blocks. For nodes with inconsistent messages, credit points are deducted and node behavior is recorded. If the messages sent by more than F consensus nodes are inconsistent with the messages of the PN, the PN will terminate the current round of consensus, deduct the reputation points of the consensus nodes with inconsistent messages, and restart the consensus process.

4.4. View change protocol

The improved view change protocol selects the PN from the excellent nodes in the system, not from all nodes. The remaining excellent nodes will be used as candidate nodes to take over the role of the PN when the PN fails. According to the credit score, the classified excellent nodes in the system are numbered as $0, 1, \ldots, |excel| - 1$ according to the score. The selection probability of the PN is related to its number. The smaller the number, the greater the probability that the node becomes the PN.

$$prim = view \bmod |excel| \tag{6}$$

where $prim$ denotes the sequence number of the finally selected PN. $view$ denotes the number of the current view and $|excel|$ denotes the

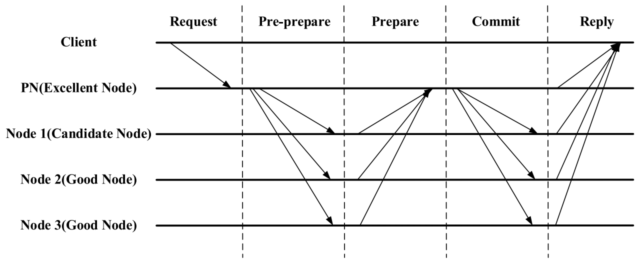

Fig. 6. Consistency protocol of CE-PBFT.

number of all excellent nodes. When the PN does not receive $2F + 1$ pieces of preparation information within the fixed time t_1, the improved view change protocol will be executed, where F denotes the maximum number of nodes allowed to fail in all nodes. In addition, when the blocks fail to be generated within the specified time t_2, the improved view change protocol will also be triggered, where $t_2 > t_1$.

4.5. Time complexity analysis

In PBFT, the number of communications in a complete consensus process is calculated as formula (7).

$$T_1 = 1 + N + N^2 + N^2 + N = 2N^2 + 2N + 1 \tag{7}$$

where N denotes the number of nodes. However, in the improved CE-PBFT, since the consistency protocol is simplified and the decision tree classification process is not considered, the number of communications in the three stages is N, so it can be expressed as $T_2 = 4N$. It can be seen that $T_1 > T_2$. By ignoring the decision tree classification process, the communication complexity of CE-PBFT is reduced from $O(N^2)$ to O(N). When the system needs to remove Byzantine nodes, considering the complexity of the ID3 decision tree algorithm, the CE-PBFT algorithm may result in higher time complexity when Byzantine nodes exist. However, due to the effective improvement in throughput, transaction latency, and communication overhead of CE-PBFT compared to other protocols, compared to the benefits of improved optimization, the increased time complexity of this part is worth it.

4.6. Correctness analysis

CAP principle is a basic principle in distributed system design, which states that three attributes cannot be satisfied at the same time in a distributed computing system: consistency, availability and partition tolerance. Consistency requires that all nodes of the system have the same data copy or view at the same time. Availability requires that the system can respond to requests at any time, that is, the system has no fault or delay. Partition tolerance requires that the system can continue to work, even if some parts of the network cannot communicate or fail. According to the principle of CAP, when a network partition occurs, to guarantee availability and tolerance, the system needs to replicate data copies between different nodes in the partition, which may lead to data inconsistency between different nodes. On the contrary, if the system requires consistency, that is, all nodes have the same data, it needs to sacrifice availability or partition tolerance.

The existence of CAP principle means that it is necessary to weigh and choose the degree of consistency, availability and partition tolerance in the design of a distributed system. Different systems may make different choices on these three attributes, and determine which attributes are the most important according to specific application requirements and business scenarios. According to the CAP theorem, the consistency protocol of CE-PBFT algorithm simplifies the process and ensures the availability. In CE-PBFT, each non-fault consensus node will send a response to its PN to ensure the availability of the algorithm. CE-PBFT eliminates the fault nodes through the decision tree classification, and when the PN fails, the PN is replaced in time through the improved view change protocol to ensure the fault tolerance of the algorithm. The consistency of CE-PBFT algorithm refers to strong consistency, that is, to achieve a consistent state through appropriate ways while meeting the availability and partition tolerance. In the CE-PBFT algorithm, each node saves the records. When a new network node is added, it synchronizes the block information through the message transmission in the consistency protocol, and finally realizes the strong consistency of the algorithm.

Table 3
Experiment setting.

Experiment setting	Configuration
Operating System	Ubuntu 18.04
Memory	16GB DDR4
CPU	Intel Core i7-8700K @ 3.70 GHz
JDK	8
Number of experiment	20
Number of message	200
Number of blockchain nodes	[100, 700]
Initial credit value	70

5. Experiment and discussion

5.1. Experiment setting

To evaluate the performance of the proposed CE-PBFT, a blockchain system is simulated and implemented in JAVA on a computer with an Intel Core i7-8700K@3.70 GHz processor and Ubuntu 18.04 operating system. CE-PBFT is compared to PBFT (Castro et al., 1999), G-PBFT (Lao et al., 2020), RBFT (Lei et al., 2018), WBFT (Qin et al., 2022) and PPoR (Abishu et al., 2021), using communication overhead, throughput, and latency as performance metrics. A series of experiments are conducted, and corresponding experimental data is collected. To simulate the transaction workload of large-scale consortium blockchain applications in traceability scenarios, a series of transaction data is created and generated. These transaction data include information from various stages such as raw material procurement, production processes, and logistics transportation. The number of nodes in the network is gradually increased from 100 to 700 for experimental testing. The initial trust value for all nodes is set to 70. After the system runs for 10 min, the client initiates 200 sets of requests. The experiments are repeated 20 times, and the average of the 20 experimental results is taken as the final result. Specific experiment settings are listed in Table 3.

In the experiment, the system's throughput is measured by recording the number of successfully completed transactions. The transactions per second (TPS) serves as the metric for measuring throughput. By continuously increasing the number of concurrent transactions and observing the number of transactions successfully processed by the system within a unit of time, the system's throughput is obtained. To measure transaction latency, the time taken from transaction initiation to confirmation and execution is recorded. In the experiment, a scenario with concurrent transactions is simulated, and the execution time of each transaction is measured. By measuring multiple transactions, the average transaction latency of the system is obtained. To measure communication overhead, the number and size of messages transmitted between nodes during the consensus process are recorded. Communication between nodes is monitored in the experiment, and the number of messages sent and received by each node is recorded. By analyzing and summarizing this data, the communication overhead in the system is determined.

5.2. Transaction latency

Transaction latency represents the time elapsed from one node initiating a consensus request to all nodes in the system reaching an agreement. It reflects the time required for consensus algorithm to process requests and reach consensus. Lower transaction latency means faster consensus process, which can obtain better system response speed.

In Fig. 7, the changes of transaction latency of PBFT, G-PBFT, RBFT, WBFT, PPoR and the proposed CE-PBFT with the increase of nodes in the network are shown. The abscissa in the figure denotes the number of network nodes, and the ordinate denotes the transaction latency. It can be seen from Fig. 7 that under the same conditions, the latency of PBFT algorithm and RBFT algorithm is higher than that of CE-PBFT algorithm. Under the condition of the same transaction processing and different number of nodes, the transaction latency will increase with the increase of the number of nodes, but the consensus latency of the improved CE-PBFT algorithm is lower than that of the PBFT algorithm. When the number of nodes is 100, the latency of the six consensus algorithms is in the range of 1 ms to 3 ms. With the increasing network nodes, the latency of PBFT and RBFT increases rapidly. When there are 700 network nodes, the latency of PBFT is 15.6, the latency of RBFT is 13.9782, and the latency of CE-PBFT is 5.654, which is 63.75% and 59.55% lower than that of PBFT and RBFT, respectively. Among all consensus algorithms, the second best performing consensus algorithm is PPOR, with a latency of 7.4476. However, the best performing CE-PBFT is reduced by 24.83%. The latency of PBFT algorithm and RBFT algorithm is dramatically higher than that of CE-PBFT, because the time complexity of the three-stage consensus algorithm of PBFT algorithm and RBFT algorithm is high, and the node consumes a lot of time in pairwise communication, which increases the consensus latency. CE-RBFT selects nodes with high reputation points to participate in the consensus according to the reputation mechanism. These consensus nodes have good stability and reliability, which is conducive to reducing errors of consensus process and effectively enhancing the efficiency of the consensus.

In Fig. 8, the results of 20 independent experiments of these six consensus algorithms in the 700 node consortium blockchain scenario are shown. Through Fig. 8, it can be clearly seen that CE-PBFT is not only smaller in value, but also smaller in curve fluctuation range. The latency of CE-PBFT fluctuates around 6 ms, while the latency of other consensus algorithms is more than 7 ms, and the latency of RBFT and PBFT is even more than 13 ms. The latency value is smaller, which further shows that the consortium blockchain using CE-PBFT has lower latency than many consensus algorithms. The curve fluctuation range is smaller, which shows that the latency results of CE-PBFT are more stable than those of other consensus algorithms.

The latency reduction of CE-PBFT can be attributed to the improvement and optimization of view change protocol. In order to prevent the Byzantine node from becoming the PN and frequent view change, which leads to the problem of low system efficiency, the view change protocol is improved to narrow the range of the PN to the excellent nodes with good node reputation, so as to ensure the efficiency and credibility of the PN selection. The low latency of CE-PBFT implementation is crucial for fast transaction acknowledgment and response. It ensures faster transaction processing and shortens the waiting time in the process of transaction verification, and improves the user experience. Through the tests, the effectiveness of CE-PBFT consensus algorithm in time latency is verified. The low latency provided by CE-PBFT makes it a critical solution for low latency transaction processing.

5.3. Throughput

To evaluate the performance of CE-PBFT in throughput, a series of experiments are conducted to compare the performance of PBFT, G-PBFT, RBFT, WBFT, PPoR and CE-PBFT under different node numbers. Throughput represents the number of consensus requests that the system can complete within a given time. It measures the processing power of the consensus algorithm and the capacity of the system. Higher throughput means that the system can handle more requests or transactions and has higher processing efficiency. Starting from the number of 100 consortium blockchain nodes, it gradually increases to 700 nodes. For each different scenario, the average throughput of six consensus algorithms in all experiments is measured.

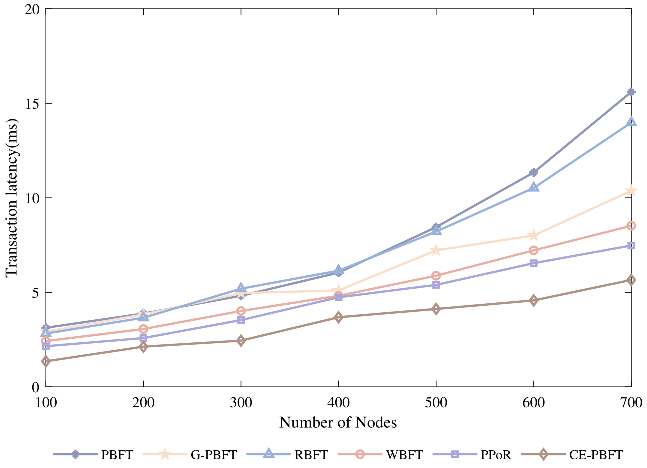

Fig. 7. The transaction latency of the six consensus algorithms.

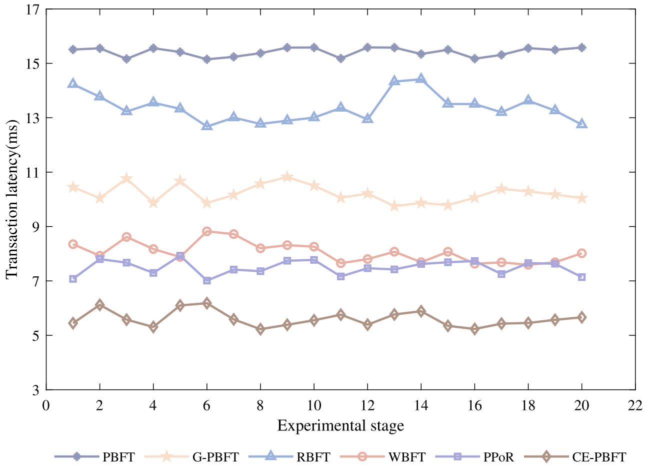

Fig. 8. The transaction latency in 20 experiments.

Fig. 9 shows the throughput changes of PBFT, G-PBFT, RBFT, WBFT, PPoR and the proposed CE-PBFT with the increase of nodes in the network. The value of abscissa in Fig. 9 denotes the number of network nodes, and the value of ordinate in the figure denotes the throughput of the network. The results prove that the throughput of CE-PBFT is vastly higher than that of other algorithms under the same experimental conditions. When there are 100 network nodes, the throughput of PBFT algorithm and CE-PBFT is close, about 800, and the throughput of other algorithms including G-PBFT, RBFT and WBFT is about 600. As network nodes gradually increase, the throughput of all algorithms is declining. The decline rate of PBFT is much higher than that of other algorithms, and finally becomes the consensus algorithm with the lowest throughput among the six consensus algorithms. When there are 700 nodes in system, the throughput of PBFT is 92, and the value of other algorithms is less than 200, but the throughput of CE-PBFT algorithm is 268. In many scenarios, the throughput of the CE-PBFT

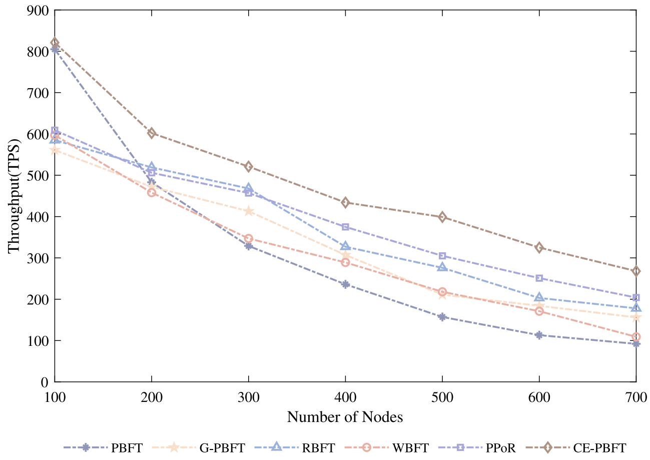

Fig. 9. The throughput of the six consensus algorithms.

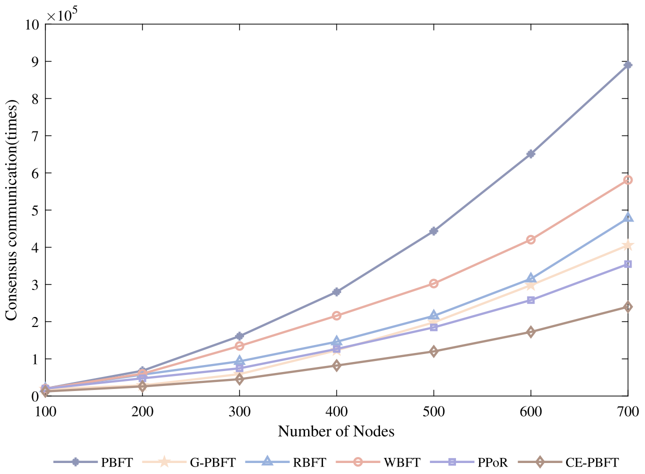

Fig. 10. The communication overload of the six consensus algorithms.

consensus algorithm is higher than that of the comparative algorithm, and the decline rate is not higher than that of comparative algorithms, which denotes that the performance of CE-PBFT algorithm is higher than that of the five comparison algorithms, and the more nodes, the more obvious its performance. This means that CE-PBFT can process transactions more efficiently and reach a higher consensus speed.

The high throughput of CE-PBFT can be attributed to the optimized reputation evaluation mechanism. By reducing redundant message exchange and improving the efficiency of message verification, CE-PBFT can process transactions faster, thus achieving higher throughput. The improved consensus protocol also reduces the complexity and further improves the throughput by reducing the time required to reach consensus.

5.4. Communication overload

In this section, the experimental results of PBFT, G-PBFT, RBFT, WBFT, PPoR and the proposed CE-PBFT in terms of communication load performance are introduced. A series of experiments are carried out to measure the communication load of these consensus algorithms in different working scenarios. In the experiment, we measure the network traffic needed in the consensus process by changing the number of consortium blockchain nodes. Communication overhead, which refers to the resources and costs required for message passing and communication between nodes in consensus algorithm, can effectively measure the efficiency of consensus algorithms. The lower communication overhead means that the consensus algorithm can communicate in a more efficient way, reducing the network load and resource consumption. Specifically, the communication overhead of different algorithms is compared by recording the number of messages delivered during the execution of the algorithm.

Fig. 10 shows the change of consensus communication overhead of six formula algorithms when there are more and more system nodes. In Fig. 10, the value of abscissa denotes the number of network nodes, and the value of ordinate denotes the consensus communication overhead. It is obvious from the experimental results that the communication overhead of CE-PBFT is relatively low under the number of each node. When there are only a few nodes, the communication load of all consensus algorithms is small. However, the communication overhead in the network increases fast as the system number of nodes grows. In terms of performance across all algorithms, the rise rate of the PBFT consensus algorithm is significantly higher than that of the comparative algorithms, while CE-PBFT has the smallest rise rate. When there are 700 nodes in the scenario, the communication overhead of CE-PBFT, G-PBFT, RBFT and PPoR are the four best performing algorithms. The communication overhead of G-PBFT algorithm is 478267, the communication overhead of RBFT is 581029, the communication overhead of PPoR is 354786, and the communication overhead of CE-PBFT is 241837, which is reduced by 49.43%, 58.38% and 31.84% respectively. This shows that CE-PBFT can reduce the amount of message passing between nodes and reduce communication overhead.

The communication load of CE-PBFT is reduced mainly because the original consistency protocol is improved and reduces the communication overhead in the practical scenario. Low communication load is very important for large-scale consortium blockchain, reducing network congestion, reducing bandwidth utilization, and ensuring more efficient use of network resources. The results of all experiments confirm the effectiveness of the communication load of CE-PBFT. The low communication load provided by CE-PBFT makes it a promising solution for applications requiring efficient use of network resources.

The experimental results demonstrate the exceptional performance of the CE-PBFT algorithm in highly dynamic or high system reliability required network environments, such as in large-scale alliances applied in traceability scenarios. The node credit evaluation model in CE-PBFT, which comprehensively considers factors such as node completion rate, consensus decay, and behavior, dynamically evaluates and reflects the reliability of nodes. This enables CE-PBFT to adapt to the frequent changes in node states in the network and maintain an efficient consensus mechanism in the ever-changing network environment. Additionally, the introduction of the node credit evaluation model and the decision tree algorithm in CE-PBFT effectively addresses the issue of Byzantine nodes and enhances the overall reliability and security of the system.

6. Conclusion

In this paper, a novel consensus algorithm CE-PBFT is designed, which classifies the network nodes by analyzing the behavior of network nodes through the designed node credit evaluation model and decision tree algorithm, and dynamically selects non-Byzantine nodes according to the classification results. It effectively reduces the number of messages and communication overload in the consensus process. Compared with PBFT, G-PBFT, RBFT, WBFT and PPoR. CE-PBFT is obviously superior to the comparison protocol in system throughput, transaction latency and communication overhead.

The proposed CE-PBFT shows good fault tolerance performance when dealing with Byzantine nodes, but it may still have some limitations for other types of faults, such as crash faults or delay faults. In the future research, we will further explore how to deal with various types of faults to improve the robustness and fault tolerance of the system.

Meanwhile, we will improve the proposed algorithm and adopt it to a wider range of practical scenarios to achieve a more reliable and efficient distributed consensus.

Declaration of competing interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Acknowledgments

This paper is funded by the project of Group intelligent autonomous operation intelligent farm, grant number 2022ZD0115800; Tech Corps Science and Technology Plan - Key Technology Research and Application of High Penetration New Energy Power Grid Dispatching Project, Youth and middleaged Scientific and Technological Innovation Leading Talents Program of the Corps, grant number 2018CB006; the Corps innovative Talents plan, grant number 2020CB001; the China Postdoctoral Science Foundation, grant number 220531; Funding Project for High Level Talents Research in Shihezi University, grant number RCZK2018C38; Project of Shihezi University, grant number ZZZC201915B; Innovative Development Project of Shihezi University, Construction and Application Demonstration of a Blockchain Based Xiangli Quality Traceability Platform, Grant number CXFZ202203; Innovation and Development Support Plan for Key Industries in Southern Xinjiang, Integration and Demonstration of Commercialization Processing Technology for Korla Fragrant Pears, Grant number 2022DB006.

References

Abishu, H.N., Seid, A.M., Yacob, Y.H., Ayall, T., Sun, G., Liu, G., 2021. Consensus mechanism for blockchain-enabled vehicle-to-vehicle energy trading in the internet of electric vehicles. IEEE Trans. Veh. Technol. 71 (1), 946–960.

Bai, Y., Hu, Q., Seo, S.-H., Kang, K., Lee, J.J., 2022. Public participation consortium blockchain for smart city governance. IEEE Internet Things J. 9 (3), 2094–2108. http://dx.doi.org/10.1109/JIOT.2021.3091151.

Boateng, G.O., Sun, G., Mensah, D.A., Doe, D.M., Ou, R., Liu, G., 2022. Consortium blockchain-based spectrum trading for network slicing in 5G RAN: A multi-agent deep reinforcement learning approach. IEEE Trans. Mob. Comput..

Cao, B., Zhang, Z., Feng, D., Zhang, S., Zhang, L., Peng, M., Li, Y., 2020. Performance analysis and comparison of PoW, PoS and DAG based blockchains. Digit. Commun. Netw. 6 (4), 480–485. http://dx.doi.org/10.1016/j.dcan.2019.12.001.

Castro, M., Liskov, B., et al., 1999. Practical byzantine fault tolerance. In: OsDI. Vol. 99, (1999), pp. 173–186.

Chen, Y., Li, M., Zhu, X., Fang, K., Ren, Q., Guo, T., Chen, X., Li, C., Zou, Z., Deng, Y., 2022a. An improved algorithm for practical byzantine fault tolerance to large-scale consortium chain. Inf. Process. Manag. 59 (2), http://dx.doi.org/10.1016/j.ipm.2022.102884.

Chen, X., Nguyen, K., Sekiya, H., 2022b. On the latency performance in private blockchain networks. IEEE Internet Things J. 9 (19), 19246–19259. http://dx.doi.org/10.1109/JIOT.2022.3165666.

Feng, Z., Luo, Q., 2020. Evaluating memory-hard proof-of-work algorithms on three processors. Proc. VLDB Endowm. 13 (6), 898–911. http://dx.doi.org/10.14778/3380750.3380759.

Ge, Z., Loghin, D., Ooi, B.C., Ruan, P., Wang, T., 2022. Hybrid blockchain database systems: Design and performance. Proc. VLDB Endow. 15 (5), 1092–1104. http://dx.doi.org/10.14778/3510397.3510406.

Guan, S., Wang, Z., Cao, Y., 2023. A novel blockchain-based model for agricultural product traceability system. IEEE Commun. Mag. 61 (8), 124–129. http://dx.doi.org/10.1109/MCOM.002.2200815.

Jindal, A., Aujla, G.S., Kumar, N., Villari, M., 2020. GUARDIAN: Blockchain-based secure demand response management in smart grid system. IEEE Trans. Serv. Comput. 13 (4), 613–624. http://dx.doi.org/10.1109/TSC.2019.2962677.

Kang, J., Xiong, Z., Niyato, D., Wang, P., Ye, D., Kim, D.I., 2019. Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks. IEEE Wirel. Commun. Lett. 8 (1), 157–160. http://dx.doi.org/10.1109/LWC.2018.2864758.

Khor, J.H., Sidorov, M., Ong, M.T., Chua, S.Y., 2023. Public blockchain-based data integrity verification for low-power IoT devices. IEEE Internet Things J. 10 (14), 13056–13064. http://dx.doi.org/10.1109/JIOT.2023.3259975.

Kumar, A., Vishwakarma, L., Das, D., 2023. R-PBFT: A secure and intelligent consensus algorithm for Internet of vehicles. Veh. Commun. 41, http://dx.doi.org/10.1016/j.vehcom.2023.100609.

Lao, L., Dai, X., Xiao, B., Guo, S., 2020. G-pbft: a location-based and scalable consensus protocol for iot-blockchain applications. In: 2020 IEEE International Parallel and Distributed Processing Symposium. IPDPS, IEEE, pp. 664–673.

Lei, K., Zhang, Q., Xu, L., Qi, Z., 2018. Reputation-based byzantine fault-tolerance for consortium blockchain. In: 2018 IEEE 24th International Conference on Parallel and Distributed Systems. ICPADS, IEEE, pp. 604–611.

Li, W., Feng, C., Zhang, L., Xu, H., Cao, B., Imran, M.A., 2021. A scalable multi-layer PBFT consensus for blockchain. IEEE Trans. Parallel Distrib. Syst. 32 (5), 1146–1160. http://dx.doi.org/10.1109/TPDS.2020.3042392.

Li, X., Lv, F., Xiang, F., Sun, Z., Sun, Z., 2020. Research on key technologies of logistics information traceability model based on consortium chain. IEEE Access 8, 69754–69762. http://dx.doi.org/10.1109/ACCESS.2020.2986220.

Li, K.-C., Shi, R.-H., 2023. A flexible and efficient privacy-preserving range query scheme for blockchain-enhanced IoT. IEEE Internet Things J. 10 (1), 720–733. http://dx.doi.org/10.1109/JIOT.2022.3203182.

Liu, J., Xie, M., Chen, S., Ma, C., Gong, Q., 2021. An improved DPoS consensus mechanism in blockchain based on PLTS for the smart autonomous multi-robot system. Inf. Sci. 575, 528–541. http://dx.doi.org/10.1016/j.ins.2021.06.046.

Misic, J., Misic, V.B., Chang, X., 2023. Design of proof-of-stake PBFT algorithm for IoT environments. IEEE Trans. Veh. Technol. 72 (2), 2497–2510. http://dx.doi.org/10.1109/TVT.2022.3213226.

Misic, J., Misic, V.B., Chang, X., Qushtom, H., 2021. Adapting PBFT for use with blockchain-enabled IoT systems. IEEE Trans. Veh. Technol. 70 (1), 33–48. http://dx.doi.org/10.1109/TVT.2020.3048291.

Misra, N.N., Dixit, Y., Al-Mallahi, A., Bhullar, M.S., Upadhyay, R., Martynenko, A., 2022. IoT, big data, and artificial intelligence in agriculture and food industry. IEEE Internet Things J. 9 (9), 6305–6324. http://dx.doi.org/10.1109/JIOT.2020.2998584.

Myrzashova, R., Alsamhi, S.H., Shvetsov, A.V., Hawbani, A., Wei, X., 2023. Blockchain meets federated learning in healthcare: A systematic review with challenges and opportunities. IEEE Internet Things J. 10 (16), 14418–14437. http://dx.doi.org/10.1109/JIOT.2023.3263598.

Nuttah, M.M., Roma, P., Lo Nigro, G., Perrone, G., 2023. Understanding blockchain applications in Industry 4.0: From information technology to manufacturing and operations management. J. Ind. Inf. Integr. 33, http://dx.doi.org/10.1016/j.jii.2023.100456.

Qin, H., Cheng, Y., Ma, X., Li, F., Abawajy, J., 2022. Weighted Byzantine Fault tolerance consensus algorithm for enhancing consortium blockchain efficiency and security. J. King Saud Univ.-Comput. Inf. Sci. 34 (10), 8370–8379.

Qushtom, H., Misic, J., Misic, V.B., Chang, X., 2023. A two-stage PBFT architecture with trust and reward incentive mechanism. IEEE Internet Things J. 10 (13), 11440–11452. http://dx.doi.org/10.1109/JIOT.2023.3243189.

Rahman, M., Azam, M.M., Chowdhury, F.S., 2021. An anonymity and interaction supported complaint platform based on blockchain technology for national and social welfare. In: Proceedings of International Conference on Electronics, Communications and Information Technology 2021. ICECIT 2021, IEEE Bangladesh Sect; Khulna Univ, http://dx.doi.org/10.1109/ICECIT54077.2021.9641269, International Conference on Electronics, Communications and Information Technology (ICECIT), Khulna Univ, Elect & Commun Engn Discipline, ELECTR NETWORK, SEP 14-16, 2021.

Saba, T., Haseeb, K., Rehman, A., Jeon, G., 2023. Blockchain-enabled intelligent IoT protocol for high-performance and secured big financial data transaction. IEEE Trans. Comput. Soc. Syst. http://dx.doi.org/10.1109/TCSS.2023.3268592.

Song, H., Zhu, N., Xue, R., He, J., Zhang, K., Wang, J., 2021. Proof-of-Contribution consensus mechanism for blockchain and its application in intellectual property protection. Inf. Process. Manag. 58 (3), http://dx.doi.org/10.1016/j.ipm.2021.102507.

Sun, G., Dai, M., Sun, J., Yu, H., 2021. Voting-based decentralized consensus design for improving the efficiency and security of consortium blockchain. IEEE Internet Things J. 8 (8), 6257–6272. http://dx.doi.org/10.1109/JIOT.2020.3029781.

Tian, Z., Li, M., Qiu, M., Sun, Y., Su, S., 2019. Block-DEF: A secure digital evidence framework using blockchain. Inform. Sci. 491, 151–165.

Tong, W., Dong, X., Shen, Y., Zhang, Y., Jiang, X., Tian, W., 2022. CHChain: Secure and parallel crowdsourcing driven by hybrid blockchain. Future Gener. Comput. Syst.- Int. J. Esci. 131, 279–291. http://dx.doi.org/10.1016/j.future.2022.01.023.

Vangala, A., Sutrala, A.K., Das, A.K., Jo, M., 2021. Smart contract-based blockchain-envisioned authentication scheme for smart farming. IEEE Internet Things J. 8 (13), 10792–10806. http://dx.doi.org/10.1109/JIOT.2021.3050676.

Viriyasitavat, W., Bi, Z., Hoonsopon, D., 2022. Blockchain technologies for interoperation of business processes in smart supply chains. J. Ind. Inf. Integr. 26, http://dx.doi.org/10.1016/j.jii.2022.100326.

Wang, M., Zhu, T., Zuo, X., Yang, M., Yu, S., Zhou, W., 2023. Differentially private crowdsourcing with the public and private blockchain. IEEE Internet Things J. 10 (10), 8918–8930. http://dx.doi.org/10.1109/JIOT.2022.3233360.

Xu, G., Bai, H., Xing, J., Luo, T., Xiong, N.N., Cheng, X., Liu, S., Zheng, X., 2022. SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent internet of vehicles. J. Parallel Distrib. Comput. 164, 1–11. http://dx.doi.org/10.1016/j.jpdc.2022.01.029.

Xu, G., Liu, Y., Khan, P.W., 2019. Improvement of the DPoS consensus mechanism in blockchain based on vague sets. IEEE Trans. Ind. Inform. 16 (6), 4252–4259.

Yang, L., 2019. The blockchain: State-of-the-art and research challenges. J. Ind. Inf. Integr. 15, 80–90. http://dx.doi.org/10.1016/j.jii.2019.04.002.

Zhang, W., Sun, G., Xu, L., Lu, Q., Ning, H., Zhang, P., Yang, S., 2022. A trustworthy safety inspection framework using performance-security balanced blockchain. IEEE Internet Things J. 9 (11), 8178–8190. http://dx.doi.org/10.1109/JIOT.2021.3121512.

Zhao, W., Jiang, C., Gao, H., Yang, S., Luo, X., 2021. Blockchain-enabled cyber-physical systems: A review. IEEE Internet Things J. 8 (6), 4023–4034. http://dx.doi.org/10.1109/JIOT.2020.3014864.