

Article

Improvement of PBFT Consensus Algorithm Based on Affinity Propagation Clustering in Intellectual Property Transaction Scenarios

Dan Du, Wenlong Feng *, Mengxing Huang, Siling Feng and Jing Wang

School of Information and Communication Engineering, Hainan University, Haikou 570228, China; 22220854000035@hainanu.edu.cn (D.D.); huangmx09@163.com (M.H.); fengsiling@hainanu.edu.cn (S.F.); 22220854000028@hainanu.edu.cn (J.W.)

* Correspondence: fwlfwl@163.com

Abstract: In response to the problems of random selection of primary nodes, high communication complexity, and low consensus efficiency in the current consensus mechanism for intellectual property transactions, a Practical Byzantine Fault Tolerance (PBFT) consensus algorithm based on the Affinity-Propagation (AP) clustering algorithm, termed AP-PBFT, is proposed. Firstly, the election strategy of the leader node is constructed based on the reputation mechanism; the reward and punishment mechanism is designed to achieve the dynamic adjustment of the reputation value of the nodes in the PBFT consensus process, and the number of votes among the nodes is introduced to determine the node's reputation value in collaboration with the reward and punishment mechanism to guarantee the precise ordering of the nodes. Secondly, nodes with high reputation values are selected as cluster centers to run the AP clustering algorithm, and clustering groups of knowledge property transaction nodes are constructed based on responsibility and availability. Finally, the three-stage consensus process of the PBFT consensus algorithm is optimized, and the consensus task is decomposed into two layers: the intra-consensus group and the inter-leader node group, reducing the communication complexity of transaction data in the blockchain. Experimental findings indicate a significant performance improvement of the algorithm over the PBFT consensus algorithm in communication complexity, throughput, and consensus efficiency in the simulation environment of multiple types of transactions in intellectual property transactions, including different types of large-scale transaction scenarios, such as purchases, sales, licenses, and transfers.

Keywords: consortium blockchain; consensus algorithm; PBFT algorithm; intellectual property transactions; AP clustering algorithm



Citation: Du, D.; Feng, W.; Huang, M.; Feng, S.; Wang, J. Improvement of PBFT Consensus Algorithm Based on Affinity Propagation Clustering in Intellectual Property Transaction Scenarios. *Electronics* **2024**, *13*, 1809. <https://doi.org/10.3390/electronics13101809>

Academic Editors: Fabio Grandi and Mehdi Sookhak

Received: 16 March 2024

Revised: 25 April 2024

Accepted: 4 May 2024

Published: 7 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the advent of the information age, the role of intellectual property has become increasingly prominent. Innovation has become the engine of economic growth, and enterprises and individuals are transforming creativity into forms of intellectual property such as patents, trademarks, and copyrights [1]. Protecting intellectual property and providing reasonable returns to holders is crucial [2]. However, the process of intellectual property transactions still faces a series of challenges, including high transaction costs, time consumption, lack of transparency, and complex legal procedures under traditional transaction models [3], severely restricting the efficient operation of the intellectual property market and hindering the adequate circulation of intellectual property and the promotion of innovation [4].

With the rapid development of blockchain technology, there is a growing recognition of its potential to address the issue of intellectual property transactions. Blockchain technology incorporates the best features of many established technologies [5,6], including cryptography, peer-to-peer networks, hashing algorithms, and consensus mechanisms [7,8].

Blockchain technology, characterized by distributed accounting and storage, traceability, prevention of information tampering, and trustlessness [9–11], can be utilized in various domains [12], including accurate recording of multiple stages of intellectual property rights such as authentication, protection, and transactions [13]. In a blockchain system, creators can record information about their works on the chain, and the system can pre-confirm the intellectual property. In the event of disputes over intellectual property infringement, users can use the data stored in the blockchain as evidence of ownership of the work, thus protecting their legitimate rights and interests.

Numerous scholars have embarked on thorough investigations into intellectual property transactions, and research programs around blockchain technology and intellectual property transactions are gradually emerging. For example, in ref. [14], the use of blockchain technology to construct a framework for intellectual property transactions provides an in-depth analysis of the application paths of multiple functional modules, such as authentication, consensus mechanisms, and on-chain supervision, in intellectual property informational transactions and puts forward a series of strategies and suggestions for promoting the development of intellectual property transactions by blockchain technology, but does not discuss in detail the challenges and limitations that may be encountered by the framework in its practical application. In ref. [15], an in-depth discussion on whether blockchain technology is used to protect intellectual property rights, such as patents and copyrights, suggests that smart contracts play a crucial role in intellectual property rights that are not registered or do not need to be registered and that blockchain technology can provide the necessary confidentiality for trade secrets. However, it lacks a thorough examination of the practical implementation of essential blockchain technologies within the realm of intellectual property rights. In ref. [16], a strategy for digital content protection and transactions leveraging blockchain and Ethereum technologies is presented. This strategy employed encryption algorithms to combat smart forgery and hacker attacks, thereby minimizing security risks in digital content transactions and enhancing transaction transparency. However, the main technology focuses on encryption algorithms and does not involve consensus mechanisms, which may be deficient in terms of consensus mechanisms when facing cyber-attacks or malicious behavior. In ref. [17], a Polkadot-based heterogeneous multi-chain model is presented, which designs a comprehensive intellectual property protection scheme for core business functions such as intellectual property registration, transactions, and rights protection. This scheme achieved the three major functions of intellectual property protection: rights confirmation, authorization, and rights protection. However, it may face challenges in practical applications such as cross-chain interaction and standardization. In ref. [18], a framework for protecting digital assets and intellectual property using blockchain technology is presented. It utilized blockchain technology to store detailed transaction information and employed watermark data to assist in tracking the transfer of asset ownership, but critical issues, such as the limitations of blockchain technology in terms of performance and scalability, are not discussed in detail.

In the exploration of smart transaction applications based on blockchain technology, e.g., [19], the principles of blockchain technology are applied to the cloud transaction system of intellectual property rights. A comprehensive coupling analysis is carried out, which proposes a strategy to optimize the trust structure and trust relationship in the transaction system. Accordingly, it constructs a game-theoretic model of the system's trust relationship. Still, the practical application may face challenges in terms of the design of the trust model and the determination of the game parameters. In ref. [20], a digital music copyright protection and trading system was developed using the Hyperledger Fabric platform to store music feature fingerprints on the Interplanetary File System (IPFS) and store the hash addresses returned by IPFS in the blockchain. This approach overcomes the limitations of blockchain scaling and storage space costs without exploring the issues of safeguarding the security and privacy of the hash addresses stored in IPFS. In ref. [21], a distributed computing resource trading system is proposed that uses multiple preference matching mechanisms to incentivize participation from collaborators holding high reputation scores

while automating decentralized matching and reputation mechanisms through blockchain technology. In ref. [22], the basic model of the blockchain system and the advantages of its application are analyzed in depth, and the innovation of a new type of intellectual property transaction system is explored to improve the efficiency of intellectual property transactions, which, although providing a valuable theoretical basis, does not address the specific implementation details of the key blockchain technology. In ref. [23], combined with blockchain technology to analyze the current situation of digital asset transactions, a technical architecture model based on the 'blockchain network' is proposed, which can improve transaction efficiency and save transaction costs.

Based on the above research analysis, ample room exists for developing intellectual property transactions combined with blockchain and consensus mechanisms. Existing blockchain consensus mechanisms applied to intellectual property are still primarily in the form of consortium chains [24].

Existing mainstream consensus mechanisms, notably the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm [25–27], face performance bottlenecks in intellectual property transaction scenarios, such as communication complexity, consensus efficiency, and random selection of primary nodes [28–30], which pose limitations. Aiming at these problems of PBFT, this paper proposes a PBFT optimization scheme based on the AP clustering algorithm [31], addressing issues within the consensus mechanism of intellectual property-centric transactions. The main contributions are as follows:

- (1) Proposed a reputation mechanism based on incentive mechanisms and the number of votes exchanged between nodes to assess and rank each node's reputation value. Selecting nodes with higher reputation values as leader nodes responsible for message propagation and broadcasting effectively reduces the likelihood of main node errors, decreases the frequency of view switches in the system, and enhances the accuracy and efficiency of consensus.
- (2) The improvement of the original AP algorithm for its initial clustering center selection is more random, combining the reputation mechanism with the AP clustering algorithm, selecting some of the high-reputation value nodes as the initial clustering center, serving as a means to cluster nodes within the network, attenuating the influence of the initial nodes in the AP algorithm on the clustering results, ensuring the authority and credibility of the results, and improving the reliability of the final generation of the block.
- (3) The optimization of the three-phase consensus process of the PBFT consensus algorithm has been carried out, achieving an improvement from global decentralization consensus to layered multi-centralized consensus. By adopting the consensus method of intra-group before inter-group, the communication frequency between nodes and the storage burden of nodes have been effectively reduced, thereby lowering the transaction cost of intellectual property rights.

For the rest of this paper, Section 2 describes the design of the AP-PBFT algorithm. Section 3 delves into an experimental assessment of the algorithm's performance. Section 4 wraps up the paper and suggests directions for improvement in future research.

2. AP-PBFT Consensus Algorithm Design

2.1. Overall Algorithm Design

The AP-PBFT algorithm comprises three components: reputation mechanisms, clustering and grouping, and optimization of three-stage communication. The algorithm's comprehensive flowchart is illustrated in Figure 1.

Firstly, the number of groups is determined, assuming that the total nodes' number is N and the number of groups is K . Node reputation's total value is influenced by both the reward-punishment mechanism and the tally of inter-nodal votes. When evaluating the behavior of a node in completing a PBFT consensus, a node with good performance will be rewarded accordingly; on the contrary, it will be punished if there is malicious behavior. In addition, each node has a voting right of 1 vote and can vote for any node except itself.

When a node makes frequent good contributions and is trusted by most of the other nodes, the reputation value increases, and selecting nodes with high reputation values as leader nodes enhances the consensus algorithm's dependability and reliability. After calculating the reputation value of the nodes, they are sorted, and each node is assigned a unique number in the range of $0, 1, \dots, K - 1, \dots, N$. Define node 0 with the highest reputation value as the primary node, nodes $1, 2, \dots$, up to node $K - 1$ with higher reputation values as leader nodes.

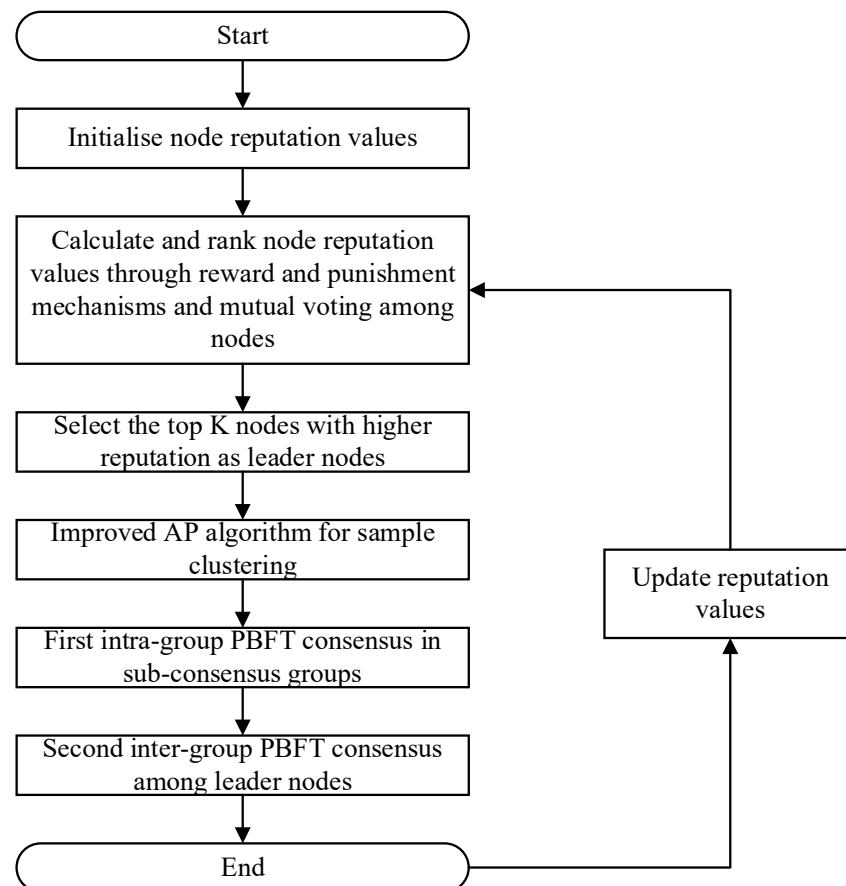


Figure 1. AP-PBFT Algorithm Flowchart.

Secondly, the leader node is used as the clustering center, and all the nodes are divided into different clusters by the improved AP clustering algorithm in order to complete the division of all the nodes, and the K groups after the clustering grouping are defined as the sub-consensus groups. After the nodes are divided into different cluster groups, a cluster is a subgroup, and each node in the group saves a list that shows the addresses of the nodes in the group, indicating that these nodes belong to the same group.

Finally, unlike the traditional PBFT algorithm, in which each node needs to communicate with all other nodes in order to reach consensus, AP-PBFT adopts intra-group and then inter-group consensus after grouping through improved AP clustering. Nodes within the sub-consensus groups first perform PBFT consensus in parallel, and within each sub-consensus group, the leadership node is assumed by the node with the highest reputation value, tasked with broadcasting and relaying intra-group messages, gathering intra-group consensus outcomes, and relaying them to the node boasting the highest inter-group reputation value, known as the global primary node. Nodes within each subgroup only need to carry out internal consensus and store the transaction data within the group without the need to back up all transaction records within the entire system, thus significantly reducing the communication count between nodes and the storage burden on nodes. Then, each leader node, including the primary node, starts inter-group consensus confirmation,

and once the consensus is reached, the final result will be returned to the client. When PBFT consensus is performed between the leader nodes, the consensus process is more efficient due to the fact that a partial consensus has already taken place, and the scope of the consensus is smaller.

In intellectual property transaction scenarios, creators can upload their original works to the blockchain. Buyers can query information based on their needs. Successful arrangements between buyers and sellers in intellectual property transactions lead to the establishment of a ledger based on specific transaction details. Requests for transaction consensus are generated through contracts and transmitted to the blockchain network. Nodes participating in consensus reach an agreement on the transaction and provide feedback to the parties involved.

2.2. Election of Leader Nodes Based on Reputation Mechanism

The traditional PBFT algorithm does not directly consider the reputation of the nodes; every blockchain node has the chance to operate as the leader node, resulting in some uncertainty in node selection. In order to effectively increase the difficulty of Byzantine nodes becoming the primary nodes and prevent fraud, the reputation mechanism is designed in conjunction with the process of intellectual property transactions.

Assuming that each node has an initial reputation value, the node's reputation value becomes refreshed upon the successful completion of the PBFT consensus task in the blockchain network before the node clustering, and the updated node's reputation is determined by the combination of the initial reputation value, the rewards received by the node, the penalties imposed on the node, and the vote count received by the node.

For nodes that frequently participate in IP transactions and show good behavior, including submitting correct messages on time during the completion of a PBFT consensus, honestly verifying messages from other nodes and making correct decisions, actively participating in the consensus process, contributing effective computational resources, etc., they can be given additional rewards to enhance their reputation value to motivate them to continue to participate and maintain a high level of reputation. The reward mechanism is designed based on a sigmoid function, which makes the reward value obtained by a node larger when its reputation value is low, but the growth rate of the reward gradually slows down with the increase in the reputation value, preventing malicious nodes from rapidly increasing their reputation value through one-time high rewards. The weighted sum of the reward values obtained by the node's good behavior is taken as input to obtain the final reward value of the node, as in Equation (1).

$$R_{new} = \frac{r_{max}}{1 + e^{-\alpha(\sum_{i=1}^n w_i \times r_i)}} \quad (1)$$

where w_i is the weight of the i^{th} good behavior of the node, n is the number of good behaviors of the node, r_i is the reward value obtained by the i^{th} good behavior of the node, r_{max} is the maximum reward value that the node can obtain.

For nodes with malicious behaviors or irregularities, including nodes sending wrong, forged, or incomplete messages during the process of completing a PBFT consensus, intentionally refusing to participate in the consensus process or not participating as required, malicious denial-of-service attacks, and so on, appropriate punitive measures should be taken to reduce their reputation scores and motivate the nodes to comply with the rules and safeguard the overall interests of the system. The punishment mechanism ensures that even if a node performs well in some aspects, it will still be punished if it performs poorly in other aspects by maximizing the node's performance on malicious behavior. The value of the penalty received is calculated through Equation (2).

$$P_{new} = \max(p_{error}, p_{reject}, p_{malicious}) \quad (2)$$

where p_{error} is the penalty value suffered by the node for sending wrong, forged, or incomplete messages, p_{reject} is the penalty value that a node receives for intentionally refusing to participate in the consensus process or for not participating as required, and $p_{malicious}$ is the penalty value suffered by the node for a malicious denial of service attack.

Choosing the leader node by considering only the node reputation value, the leader node exists only in the part of nodes with high reputation value, which can cause monopoly of high reputation value nodes and inertia of common nodes. Once the high reputation value nodes are fraudulent, the transaction data may be tampered with or forged. Therefore, the number of votes among nodes should be taken into consideration. Each node is mandated to have one vote and is free to choose the node it trusts to vote for but may not vote for itself to increase the chance of success in reaching a consensus. The total number of votes for node i is equal to the sum of all the nodes that voted for node i . The number of votes is calculated through Equation (3).

$$V_x = \sum_{y=0}^N S_y \quad (3)$$

where S_y is the voting of node y for node x . If node y votes for node x , then $S_y = 1$, otherwise $S_y = 0$.

The node's reputation is determined by considering both the node reward and punishment mechanism and inter-node voting, as in Equation (4):

$$C_{new} = C_{init} + w_1 \times \sum_i^j R_{new} - w_2 \times \sum_i^j P_{new} + w_3 \times V_x \quad (4)$$

where w_1 is the weight of the reward mechanism, w_2 is the weight of the penalty mechanism and w_3 is the weight of the node's received votes.

The node reputation value undergoes updating and sorting in each consensus cycle; the leader node is chosen according to its greater reputation value, and the primary node is selected based on its highest reputation value. If a node's reputation value falls below the initial reputation threshold, it is determined to be a malicious node and rejected.

2.3. Grouping Based on Improved AP Clustering Algorithm

Based on the types of intellectual property transactions, consensus nodes are divided into patent transaction groups, domain name transaction groups, trademark transaction groups, copyright transaction groups, and other types of intellectual property transaction groups using an improved Affinity Propagation (AP) clustering algorithm. The task of every consensus group for transactions is to verify the corresponding types of intellectual property transaction requests from clients and convey the verification outcomes to the primary node in the consensus group. For instance, the patent transaction group is responsible for verifying patent-related transaction requests, including operations such as purchasing and transferring patents; the domain name transaction group verifies domain name-related transaction requests, such as internet domain names; and the trademark transaction group verifies trademark-related transaction requests, including registration and licensing of trademarks. Grouping allows for more effective organization and management of the intellectual property transaction system. Each transaction consensus group works independently, processing multiple transaction requests in parallel with other groups, thereby improving system processing capacity and throughput.

The selection of AP clustering centers significantly influences consensus performance and implementation. Therefore, improvements need to be made to the AP algorithm. This paper focuses on two aspects of improvement. The original AP algorithm takes the median of all numbers on the main diagonal of the similarity matrix. The higher the reference degree, the greater the likelihood that the data point will become the final clustering center. This paper constructs a credibility mechanism, calculates the credibility values of each

node, and sorts them. The top K nodes with higher credibility values are selected, and their reference degrees are, respectively, assigned in order, making the top K nodes with credibility values the clustering centers, thus reducing the impact of the AP algorithm on the initial nodes. Additionally, each grouping size post-AP clustering is random. Therefore, a judgment condition is added to ensure that each group contains no less than four nodes and that the number of groups is no less than four.

Assuming there are N nodes and a data object matrix composed of M attributes, clustering primarily hinges on N nodes' similarity during iterations. The similarity matrix $s(i, k)$ is generally represented by the negative value of Euclidean distance. The larger this value, the higher the similarity between two data points, denoted as the Euclidean distance $d(x_i, x_j)$ between data objects x_i and x_j , calculated through Equation (5).

$$d(x_i, x_j) = \sqrt{\sum_{w=1}^M (|x_{iw} - x_{jw}|_2)} \quad (5)$$

According to Equation (6), the similarity matrix $s(i, k)$ is calculated, where the values on the main diagonal of the similarity matrix are called the preference, representing the reference degree of consensus node i as a clustering center. It can also be denoted as $p(i)$ or $s(i, i)$. The higher the preference, the higher the likelihood that consensus node i becomes a clustering center.

$$s(i, j) = \begin{pmatrix} p(i) & \cdots & d_{1N} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & p(i) \end{pmatrix} \quad (6)$$

Clustering is achieved by iteratively passing two types of messages between the given data points: attraction information (responsibility) and affiliation information (availability). Each iteration updates attraction and affiliation information by adding a damping coefficient λ to reduce the oscillations produced during the AP clustering algorithm iteration process. Responsibility $r(i, k)$ indicates the degree to which point k is suitable as the clustering center for data point i , $r(i, k)$ and $a(i, k)$ are initialized as zero matrices. Responsibility is calculated according to Equation (7), and Equation (8) iteratively updates responsibility.

$$r(i, k) = \begin{cases} s(i, k) - \max_{j \neq k} \{a(i, j) + s(i, j)\}, & i \neq k \\ s(i, k) - \max_{j \neq k} \{s(i, k)\}, & i = k \end{cases} \quad (7)$$

$$r_{t+1}(i, k) = \lambda r_t(i, k) + (1 - \lambda) r_t(i, k) \quad (8)$$

Availability $a(i, k)$ represents the fitness of consensus node i to choose point k as the clustering center, calculated according to Equation (9) and iteratively updated according to Equation (10).

$$a(i, k) = \begin{cases} \min\{0, r(k, k) + \sum_{j \neq i, k} \max\{r(j, k), 0\}\}, & i \neq k \\ \sum_{j \neq k} \max\{r(j, k), 0\}, & i = k \end{cases} \quad (9)$$

$$a_{t+1}(i, k) = \lambda a_t(i, k) + (1 - \lambda) a_t(i, k) \quad (10)$$

Upon reaching the predefined parameter M, the AP clustering algorithm concludes its iterations. Otherwise, attraction and affiliation are recalculated, and then iterative updates are performed.

The improved algorithm design process is shown in Algorithm 1.

Algorithm 1. Improved AP clustering algorithm grouping process

Input:

Dataset with N nodes

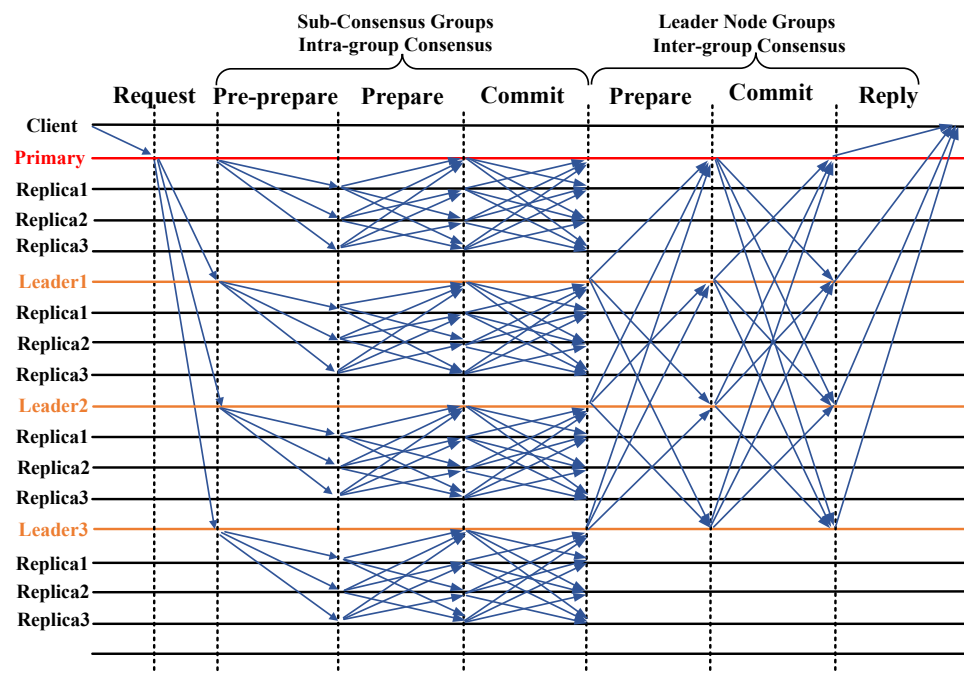
Output:

K groups with nodes of a higher reputation as cluster centers

- 1: Calculate node reputations through a reputation mechanism and sort them, assign reference $p(i)$ degrees to the top K nodes with higher reputation values in order;
- 2: Initialize the AP clustering algorithm, calculate the Euclidean distance between any nodes i and j according to Equation (5), calculate the similarity matrix between any nodes i and j according to Equation (6);
- 3: Calculate the responsibility $r(i,k)$ between any node i and cluster center k according to Equation (7);
- 4: Calculate the availability $a(i,k)$ between any node i and cluster center k according to Equation (9);
- 5: Preset a damping factor, update the responsibility $r(i,k)$ according to Equation (7) and Equation (8), and update the availability $a(i,k)$ according to Equation (9) and Equation (10);
- 6: Check node count per group. If it exceeds the maximum capacity of the group, assign the newly applied nodes to the smallest group other than the cluster center; if the number of nodes is less than 4, update the responsibility and availability values for each point and repeat this process continuously to ensure that all groups meet Byzantine fault tolerance;
- 7: If the iteration count exceeds the designated maximum or the cluster centers remain static, cease the calculation. Output the grouping result. Otherwise, return to step 2 to continue iterative calculation.

2.4. AP-PBFT Algorithm Specific Consensus Process

The number of groups obtained after completion is K by improving the AP clustering algorithm to cluster the participating consensus nodes in intellectual property transactions. The system is divided into two layers, and the sub-consensus groups formed by the clustering of each leader node's region complete the first consensus. The leading nodes with the highest credibility values in each sub-consensus group participate in the consensus, transmitting and broadcasting messages between the groups and completing the second consensus. The consensus process is shown in Figure 2.

**Figure 2.** Optimized PBFT algorithm consensus process.

2.4.1. Consensus within Sub-Consensus Groups

Pre-Prepare Phase: The algorithm enters the pre-prepare phase after the client requests the primary node. The primary node sends messages to the leader nodes of each sub-consensus group; simultaneously, the primary node sends messages to other slave nodes in its sub-group. The node's credibility value determines the primary node and each sub-consensus group's leader node. Afterward, the leader node transmits the message $\langle \text{PRE-PREPARE-SUB}, v, n, d, m, p \rangle$ to its sub-group, where v is the current view number, n is the sequence number, m is the client request message, d is the digest of m , and p is the primary node.

Prepare Phase within Sub-Consensus Groups: All nodes within the sub-consensus group receive pre-prepare messages $\langle \text{PRE-PREPARE-SUB}, v, n, d, m, p \rangle$ multicast by the leader node and proceed with inspection. Upon successful validation, each node broadcasts the message to all other nodes in the data packet. Subsequently, nodes send prepare messages $\langle \text{PREPARE-SUB}, v, n, d, m, p \rangle$ to denote their commencement of the preparation phase and log these messages locally. Once the message is prepared, the node proceeds to the next phase.

Submission Phase within Sub-Consensus Groups: When the nodes in the sub-consensus group check that the prepared messages are consistent, they enter the submission phase. In the submission phase, messages $\langle \text{COMMITTED-SUB}, v, n, d, m, p \rangle$ are sent to all other nodes, and it is checked whether they are consistent. If a node has received confirmation messages, including itself, from $2f+1$ identical data packets, the block of the sub-consensus group is written to disk.

2.4.2. Consensus between Leader Node Groups

After each sub-consensus group completes the low-order consensus, the results are sent to the leader nodes of this group, and then each leader node performs the second consensus.

Inter-Group Prepare Phase: The leader nodes of each sub-consensus group will receive data packets $\langle \text{COMMITTED-SUB}, v, n, d, m, p \rangle$ from this group and perform checks. After verification, the leader nodes will transfer the verification message to all other leader nodes. Each leader node sends messages $\langle \text{PREPARE-LEADER}, v, n, d, m, p \rangle$ to enter the preparation phase.

Inter-Group Submission Phase: When each leader node checks that the prepared messages $\langle \text{PREPARE-LEADER}, v, n, d, m, p \rangle$ are consistent, they enter the submission phase. In the submission phase, data packets $\langle \text{COMMITTED-LEADER}, v, n, d, m, p \rangle$ are sent to all other nodes, and it is checked whether they are consistent. If a leader node has received verification-consistent messages, including itself from $2f+1$, they enter the reply phase, where f is the maximum number of Byzantine nodes between groups.

Reply Phase: The leading node conducts a second consensus and returns the ultimate voting results within the region to the client, ensuring consensus among all nodes in the intellectual property network during information exchange. The PBFT algorithm allows for a maximum of one-third of nodes in the network to fail. Clients evaluate whether the current block should be written to the blockchain by monitoring the number of messages sent and received. Once an ample number of nodes have confirmed the transaction and responded with corresponding messages, the client writes the transaction to the blockchain, ensuring that all transactions undergo adequate validation and confirmation, thus preventing malicious nodes from tampering with or fabricating transactions. After a block successfully joins the blockchain, the primary node sends a message to the leading node confirming the successful addition of the new block to the chain, which is then broadcasted within each group by the leading node to update the status of nodes across the network.

3. Analysis and Experiments

In this section, the communication complexity, latency, and throughput of AP-PBFT and traditional PBFT are compared and analyzed through simulation experiments. The network environment is excluded, and all consensus nodes operate on a single host to

ensure experiment precision. The experiment utilizes Python to simulate the underlying aspects of the physical network conditions. The server operating system is Windows 10, and the experiment configuration is an Intel (R) Core (TM) i7-8565U CPU @ 1.80GHz.

3.1. Communication Complexity Analysis

Communication complexity is a measure of algorithmic efficiency. In this study, we propose clustering knowledge transaction nodes and improving globally decentralized consensus to layered multi-centralized consensus to tackle the challenges posed by the high communication complexity and poor scalability of PBFT. Table 1 shows the communication complexity of completing one round of consensus using PBFT and AP-PBFT algorithms. Assuming there are N consensus nodes in the knowledge transaction consortium chain, divided into K groups through AP clustering, with each subgroup consisting of N/K nodes, the communication complexity analysis of the AP-PBFT algorithm is as follows.

Table 1. Comparison of Communication Complexity between PBFT and AP-PBFT Algorithms.

Communication Complexities		Pre-prepare	Prepare	Commit	Total
AP-PBFT	PBFT	$N-1$	$(N-1)(N-1)$	$N(N-1)$	$2N(N-1)$
	Intra-group consensus in sub-consensus groups	$K(N/K-1)$	$K(N/K)(N/K-1)$	$N(N/K-1)$	$2N(N/K-1)+2K(K-1)$
	Inter-group consensus among leader node groups	$K-1$	$(K-1)(K-1)$	$2K(K-1)$	

In the consensus phase within each subgroup, the primary node handles client requests, verifies and orders them, and enters the pre-prepare phase. In the consensus phase of the subgroup, the primary node sends a pre-prepared message to each subgroup's leader node and then the leader node passes the message to the slave nodes in the subgroup. At the same time, the primary node sends pre-preparation messages directly to other slave nodes within its sub-group. This process, until the completion of the pre-preparation phase, has a communication complexity of $(N/K-1)$.

After receiving and verifying the pre-preparation message, the slave node sends the preparation message to all nodes in the subgroup except itself if the verification result is true. The communication complexity involved in this process is $(N/K-1)(N/K-1)$. Subsequently, upon receiving and verifying prepared messages from other nodes within the subgroup, if validation is successful, the slave nodes broadcast commit messages to all subgroup nodes, excluding themselves. Their communication complexity is $N/K(N/K-1)$. Since there are K subgroups within the consensus network, the communication complexity within each subgroup during the consensus phase is $W = 2N(N/K-1)$.

The inter-group consensus phase among leader nodes, including the primary node, is conducted for PBFT consensus by a total of K leader nodes. The communication complexity for the first phase is $K-1$, the second phase is $(K-1)(K-1)$, and the third phase is $K(K-1)$, leading to a total communication complexity of $2K(K-1)$ for the inter-leader node consensus phase.

The improved consensus algorithm's communication complexity for completing one round of consensus is the total communication complexities of both stages combined, $2N(N/K-1) + 2K(K-1)$.

The communication complexity under different numbers of nodes is shown in Figure 3. In the intellectual property transaction network, as the number of nodes increases, the communication complexity of both algorithms gradually rises. However, with an equal number of nodes, the AP-PBFT algorithm boasts reduced communication complexity compared to PBFT.

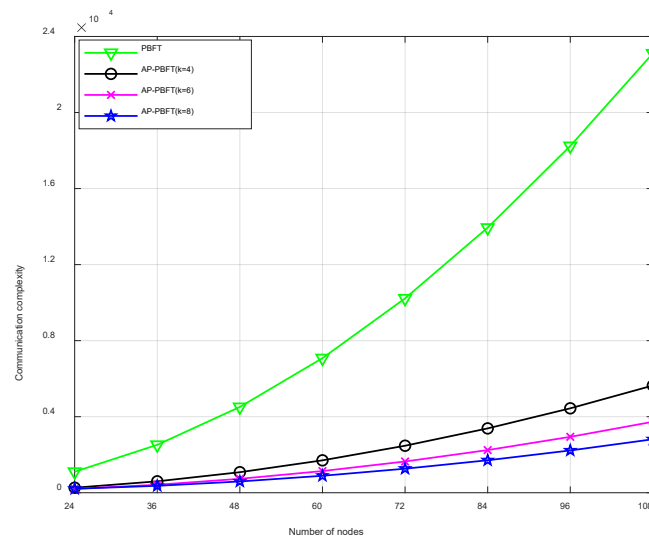


Figure 3. Comparison of communication complexity under different numbers of nodes.

To illustrate the differences more clearly, let R represent the ratio of single-consensus communication complexity between the AP-PBFT algorithm and the PBFT algorithm, as shown in Equation (11). The surface wireframe diagram is shown in Figure 4, when $K = 1$, $R = 1$, indicating that the communication complexity of AP-PBFT is the same as that of PBFT. As K increases, R initially increases to a peak before decreasing. With increasing K , the grouping count in the knowledge transaction environment rises, leading to a decrease in the number of nodes allocated to each subgroup, effectively reducing the overall workload required for consensus, thus causing R to initially increase. When K reaches a peak value, R also reaches a peak. As the number of nodes within each subgroup increases, the communication complexity of consensus significantly increases, leading to a gradual decrease in R . When $K = N$, $R = 1$, suggesting that the communication complexity of AP-PBFT equals that of PBFT. In summary, although the value of K affects the ratio of communication complexity, overall, the communication complexity of AP-PBFT significantly undercuts that of PBFT.

$$R = \frac{2N(N-1)}{2N(N/K-1) + 2K(K-1)} \quad (11)$$

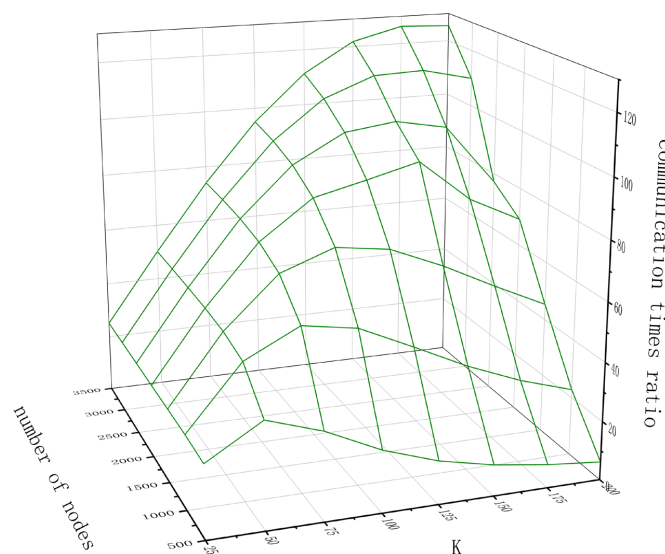


Figure 4. Surface wireframe of communication complexity ratio.

3.2. Latency Analysis

Lower consensus latency allows transactions to be confirmed quickly, improving consensus efficiency. In this study, we compare the efficiency of two algorithms through consensus latency; varying transaction volumes are simulated with identical consensus node counts; and final confirmation times are measured.

When the network nodes are 36, 48, 60, 72, 84, 96, and 108, the latency test results are shown in Figure 4. Under the same network conditions and with the same size of transaction information and data transmission, both algorithms show a trend of gradually increasing latency with an increase in node number. However, PBFT latency experiences a steep increase as the number of nodes rises. In contrast, due to the clustering grouping method adopted by AP-PBFT for nodes participating in consensus within the network, dividing the nodes into K subgroups, and improving from globally decentralized consensus to intra-group consensus and inter-group consensus, the required consensus latency is less than the original PBFT algorithm. At the same time, under the premise of the same number of nodes, when $K = 4, 8$, or 12 , the latency decreases sequentially, and the rate of latency decreases accordingly. From Figure 5, it can be seen that before K reaches its peak value, with the number of nodes N unchanged, the larger the value of K , the fewer communication times the entire consensus process requires, resulting in shorter latency.

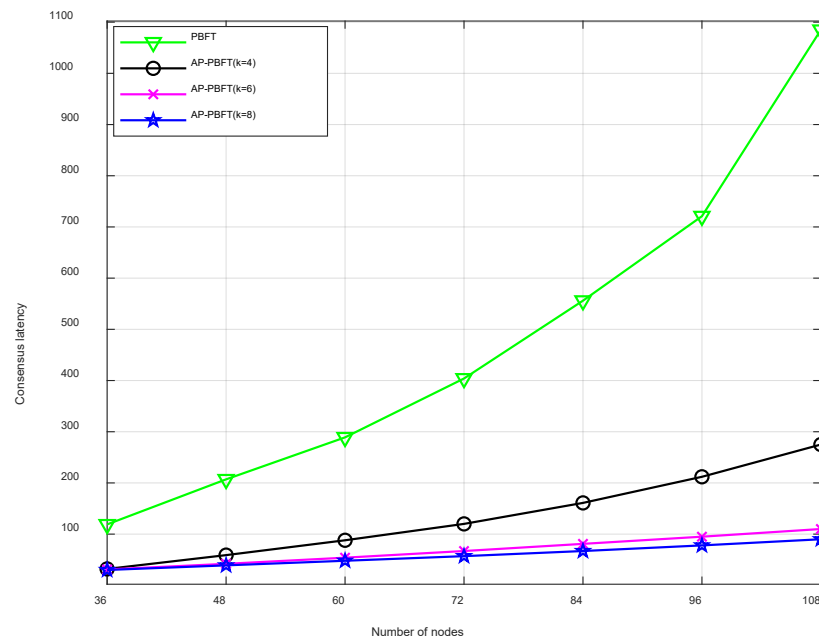


Figure 5. Delay comparison.

3.3. Throughput Analysis

As the number of nodes within each block grows, the processing time for nodes to reach consensus is expected to rise, consequently augmenting the network's load capacity. The algorithm's throughput diminishes as the node count increases. As shown in Figure 6, under the same experimental conditions, with the increase in node count, there is notably higher throughput for the AP-PBFT algorithm compared to PBFT, and this trend continues with increasing transaction volume. The throughput will grow steadily and significantly. This is because the entire network node is divided into several subsets. Meanwhile, comparative experiments with $K = 4, 8$, or 12 are conducted when the number of nodes remains unchanged. As shown in Figure 6, before K reaches its peak value, the throughput increases with the increase in K .

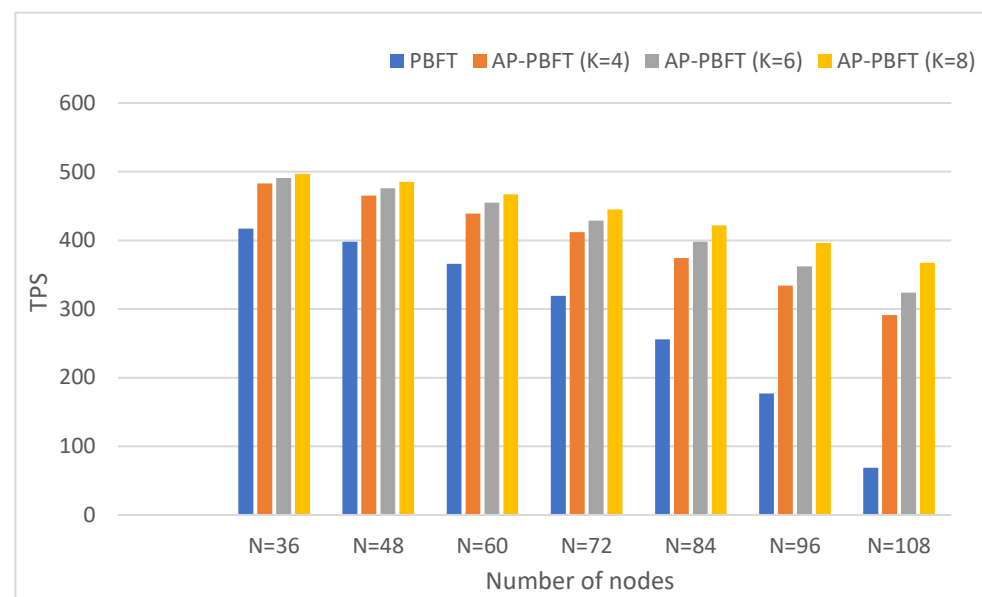


Figure 6. Throughput comparison.

4. Conclusions and Outlook

This paper presents a refined approach to optimizing the PBFT algorithm by integrating the AP clustering algorithm. It assesses node performance within the PBFT consensus process through a system of incentives and penalties. Moreover, it incorporates the voting behavior of nodes to adjust their reputation values collaboratively, subsequently sorting nodes based on their reputation levels. The nodes with higher reputation values are selected as the leader nodes of the grouping for AP clustering grouping, and sequential consensus within the sub-consensus group and between the leader node groups is used to decompose the global consensus process into a local consensus process, which reduces the communication complexity and consensus delay. Within the sub-consensus group, the nodes can perform consensus in parallel, which improves the efficiency of consensus; when consensus is achieved among leader nodes, the consensus scope is smaller, and the consensus process is more efficient because local consensus has been performed. The AP clustering grouping method reduces the burden on each node and improves the stability of the system. The experimental findings indicate that AP-PBFT exhibits better communication complexity performance, consensus delay, and throughput compared to the PBFT algorithm.

However, in practical application scenarios, with the widespread popularity of IPR transactions, the scale of nodes in the network is increasing, and more participants are joining the consensus network, resulting in a rise in the complexity of network communication and computational burden, as well as an increase in the difficulty of coordination and management among the nodes in the network. Simultaneously, with the emergence of new forms of intellectual property, more complex ownership structures, rules of use, and forms of transactions may be presented, which brings new challenges to the risk control and legal regulation of transactions. All these factors require a more robust mechanism to guarantee collaboration between nodes to ensure the overall system's stability and reliability. Therefore, the AP-PBFT algorithm still needs to be improved, and we will work on solving these problems in future work to improve its adaptability and robustness.

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

Funding: This work was supported by the Key R&D Project of Hainan Province under Grant ZDYF2024GXJS024, and the Funding for Academician Team Innovation Center in Hainan Province.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Acknowledgments: The authors would like to thank the editors and the reviewers for their valuable time and constructive comments.

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

References

- Lai, L.; Li, Y. Empowering Blockchain: Enterprise Intellectual Property Protection under the Paradigm of Networked Open Innovation. *Sci. Technol. Manag. Res.* **2023**, *43*, 213–221.
- Shang, M. Research on Improvement Strategies for Intellectual Property Rights Protection Issues in Liaoning Free Trade Zone. In Proceedings of the 20th Annual Academic Conference of the CPC Shenyang Municipal Committee and the People's Government of Shenyang, Shenyang, China, 24–26 February 2023; Volume 5. Collection of Papers of the 20th Annual Shenyang Scientific and Academic Conference—Third Prize Paper.
- Sun, X. Origin and Avoidance of Risks in Intellectual Property Cloud Transactions. *People's Forum* **2020**, *23*, 116–117.
- Li, X. Research and Implementation of Intellectual Property Transaction Platform Based on Blockchain. Master's Thesis, Henan University of Technology, Zhengzhou, China, 2023.
- Zhu, J.; Zhang, Q.; Gao, S. Research Progress on Key Technologies and Applications of Blockchain. *J. Taiyuan Univ. Technol.* **2020**, *51*, 321–330.
- Misic, J.; Misic, V.B.; Chang, X.L.; Qushtom, H. Adapting PBFT for use with blockchain-enabled IoT systems. *IEEE Trans. Veh. Technol.* **2021**, *70*, 33–48. [\[CrossRef\]](#)
- Zeng, S.; Huo, R.; Huang, T.; Liu, J.; Wang, S.; Feng, W. Review of Blockchain Technology: Principles, Progress, and Applications. *J. Commun.* **2020**, *41*, 134–151.
- He, Y. Research on Patent Protection and Transaction System Based on Blockchain Technology. Master's Thesis, Nanjing University of Posts and Telecommunications, Nanjing, China, 2023. [\[CrossRef\]](#)
- Huang, J. Research on Optimization of Internal Control in Hospitals Based on Blockchain Technology. *Econ. Obs. Bohai Sea* **2024**, *2*, 150–153. [\[CrossRef\]](#)
- Mačiulienė, M.; Skaržauskienė, A. Conceptualizing blockchain-based value co-creation: A service science perspective. *Syst. Res. Behav. Sci.* **2021**, *38*, 330–341. [\[CrossRef\]](#)
- Wang, Q.; Li, F.; Ni, X.; Xia, L.; Wang, Z.; Liang, G. Research on Blockchain Consensus Algorithm and Its Application. *J. Front. Comput. Sci. Technol.* **2022**, *16*, 1214–1242.
- Meng, B.; Wang, Y.; Zhao, C.; Wang, D.; Ma, B. Review of Blockchain Cross-Chain Protocols. *Comput. Sci. Explor.* **2022**, *16*, 2177–2192.
- Liu, Y. Research and Implementation of Intellectual Property Asset Management and Transaction System Based on Blockchain. Ph.D. Thesis, Xidian University, Beijing, China, 2021. [\[CrossRef\]](#)
- Zhou, Y.; Zhao, Y.; Chen, H. Exploration and Mechanism Design of Intellectual Property Trading Based on Blockchain. *Econ. Syst. Reform* **2021**, *06*, 112–119.
- Hauck, R. Blockchain, Smart Contracts and Intellectual Property. Using distributed ledger technology to protect, license and enforce intellectual property rights. *Leg. Issues Digit. Age* **2021**, *1*, 17–41. [\[CrossRef\]](#)
- Khan, U.; An, Z.Y.; Imran, A. A blockchain ethereum technology-enabled digital content: Development of trading and sharing economy data. *IEEE Access* **2020**, *8*, 217045–217056. [\[CrossRef\]](#)
- Ning, M.; Liu, D. Research on Digital Intellectual Property Protection Scheme Based on Blockchain Technology. *Inf. Sci. Theory Pract.* **2020**, *43*, 144–150+201.
- Bhadauria, S.; Kumar, P.; Mohanty, T. Intellectual Property Protection using Blockchain and Digital Watermarking. In Proceedings of the 2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Hyderabad, India, 13–16 December 2021; IEEE: New York, NY, USA, 2021; pp. 1–6.
- Xia, Y.; Sheng, G. Analysis of Trust Relationship in Intellectual Property Cloud Transaction System Based on Blockchain Principles. *Financ. Account. Mon.* **2021**, *19*, 116–122. [\[CrossRef\]](#)
- Niu, X.; Han, D.; Sun, Z. Music Copyright Protection and Transaction System Based on Consortium Blockchain. *Comput. Appl. Res.* **2022**, *39*, 18–23.
- Wang, T.; Ai, S.; Cao, J. A Blockchain-Based Distributed Computational Resource Trading System for Industrial Internet of Things Considering Multiple Preferences. *arXiv* **2022**, arXiv:2201.09539.
- Cheng, J. Research on Innovative System of Intellectual Property Transaction—Concept of “Blockchain Intellectual Property Trading Platform”. *Technol. Mark.* **2020**, *27*, 85–86.
- Yan, Z.; Li, J. Innovative Application of Blockchain Technology in Digital Asset Trading Field. *Enterp. Econ.* **2020**, *39*, 56–64.

24. He, J.; Zhang, K.; Xue, R.; Zhu, N.; He, P.; Son, H. A High-Reliability Blockchain Consensus Mechanism Based on Contribution Value and Difficulty Value. *Chin. J. Comput.* **2021**, *44*, 162–176.
25. Wang, X. Efficient Practical Byzantine Fault Tolerance Algorithm. *Comput. Digit. Eng.* **2023**, *51*, 997–1000.
26. Xu, G.; Bai, H.; Xing, J.; Luo, T.; Xiong, N.N.; Cheng, X.; Liu, S.; Zheng, X. SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. *J. Parallel Distrib. Comput.* **2022**, *164*, 1–11. [[CrossRef](#)]
27. Wang, X.; Fu, J.; Wang, Y. Research on Sharding Technology Based on PBFT Algorithm. *Comput. Digit. Eng.* **2024**, *52*, 213–218+246.
28. Chen, S.; Wang, B.; Chen, Y.; Xing, T.; Ma, Y.; Zhao, J. Improved PBFT Consensus Algorithm Based on Node Group Reputation Model. *Comput. Appl. Res.* **2023**, *40*, 2916–2921. [[CrossRef](#)]
29. Kumar, A.; Vishwakarma, L.; Das, D. R-PBFT: A secure and intelligent consensus algorithm for Internet of vehicles. *Veh. Commun.* **2023**, *41*, 100609. [[CrossRef](#)]
30. Xiao, J.; Luo, T.; Li, C.; Zhou, J.; Li, Z. CE-PBFT: A high availability consensus algorithm for large-scale consortium blockchain. *J. King Saud Univ.-Comput. Inf. Sci.* **2024**, *36*, 101957. [[CrossRef](#)]
31. Xiaoyong, L.; Hui, F. A Fast AP Clustering Algorithm. *J. Shandong Univ. (Eng. Sci.)* **2011**, *41*, 20–23.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.