Power trading Raft consensus mechanism considering green certificate and carbon emission weights

Xiaoping Xiong[1], Yinzheng Liu[1]*, Huijie Qu[2] and Yiming Cai[1]

[1]School of Electrical Engineering, Guangxi University, Nanning, China, [2]School of Mathematics and Information Sciences, Nanning Normal University, Nanning, China

The consensus mechanism is a critical technology in the power trading blockchain distributed power trading platforms, which are gradually being applied and promoted to achieve the "dual carbon" goal. Green-certificate trading and carbon emission trading systems help mobilize power generation enterprises to increase the generation of and investment in renewable energy, thereby becoming important factors in distributed power trading. Therefore, they should also be considered as factors in the consensus mechanism. This article first evaluates the green-certificate weight based on whether nodes possess green certificates, evaluates the carbon emission weight based on the carbon emission settlement of nodes in the previous cycle, and constructs a mechanism to evaluate a node's weight. An improved Raft consensus algorithm based on node weight is then proposed. Nodes with greater weights are more likely to become leading nodes, promoting active compliance and green consumption by nodes and reducing carbon emissions. A simulation shows that the improved Raft consensus mechanism enhances the reliability of leading nodes, improves the efficiency of the distributed power transaction, increases the generation of renewable energy, and encourages the consumption of new energy.

1 Introduction

With the proposed "carbon peak, carbon neutral" goal and the need to construct a new power system, wind power, photovoltaic power, solar energy, and other new distributed energy sources are being connected to the grid in large quantities (Li et al., 2019; Cui et al., 2020; Zhang and Kang, 2022). As a result, the energy and power trading market is becoming more decentralized, and the peer-to-peer (P2P) trading of distributed new energy sources is becoming more prevalent (Kim and Dvorkin, 2019; Tushar et al., 2020). Given the ecological advantages of new energy, China is actively promoting energy transformation (Zhang H. et al., 2023), and the system of green-certificate trading and carbon emissions trading is being gradually applied. The implementation of these two systems also affects the enterprise power generation mode and power-purchase decision-making, which affects the energy transition and the consumption of new energy. "Green certificate" is the shortened name of a green power certificate, which is an electronic voucher with a unique code identification issued for non-hydro renewable energy. Green certificates can be traded in the green-certificate market. Carbon emission rights, in contrast, allow enterprises to trade their excess carbon emission

rights on the market (Zhao et al., 2023), prompting energy savings, emission reductions, and renewable energy development. With the development and transformation of energy, some researchers have also proposed relevant research solutions to address the problems in the energy field. Guo et al. (2022) proposed a hierarchical game between distributed energy stations (DES) and hydrogen-powered vehicles (HV) under dynamic hydrogen pricing, promoting the use of renewable energy for hydrogen production and reducing the operating costs of DES. Li et al. (2020) proposed a novel Newton-Raphson algorithm to solve the double-mode energy management problem (EMP) in a fully distributed fashion, and the proposed algorithm exhibited the faster convergence feature.

With the deepening of research in the field of energy, the way of power trading is rapidly changing to distributed power trading. Unlike the traditional centralized trading method, blockchain features such as decentralization, transparency, tampering, and traceability (Meng et al., 2020) are highly compatible with distributed power trading (Tai et al., 2016; Sun et al., 2023). The blockchain method thus provides a new solution for P2P power trading (Huckle et al., 2016; Chen et al., 2018). Various projects have already applied blockchain technology to distributed P2P power transactions (Wang et al., 2016; Zhang et al., 2016; Tahir et al., 2022), and the efficient and secure consensus mechanism forms the core of blockchains for distributed power transactions (Xia et al., 2021).

Compared with conventional power trading, the proliferation of nodes brought about by the increased share of green energy sources in the distributed power trading market and the addition of a large number of distributed power trading nodes exacerbates the problems of a time-consuming, inefficient consensus process and the malicious behavior of nodes. Common consensus mechanisms include PoW (Nakamoto, 2008), PoS (King and Nadal, 2012), DPoS (Snider et al., 2018), PBFT (Castro and Liskov, 2002), Paxos (Lamport, 2019), and Raft (Ongaro and Ousterhout, 2014). PoW and PoS reside in the field of digital currencies such as Bitcoin and are highly complex, which conflicts with the requirement of real-time transmission of information on distributed power transactions. For the PBFT consensus algorithm, relevant researchers have proposed improvement plans. In the Internet of Vehicles (IoV), Xu et al. (2022) proposed the SG-PBFT consensus algorithm to address the issues of the central server of the IoV may not be powerful enough and the IoV itself may not be robust enough to single-node attacks, achieving higher consensus efficiency. Luo (2023) proposed the ULS-PBFT consensus algorithm to apply the PBFT consensus mechanism in large-scale network scenarios, which significantly reduces storage overhead and has certain advantages. The Raft consensus mechanism satisfies the requirements of transmitting information on distributed power transactions in terms of complexity and transmission speed, and other aspects satisfy the corresponding needs of distributed power transactions. However, nodes must satisfy the Byzantine fault tolerance conditions. In distributed power trading, the constraints of the power system, trading contract, and other conditions ensure that the nodes cannot modify information to meet the Byzantine fault-tolerant conditions (Fang et al., 2019). The Raft consensus mechanism can thus be used in a distributed power trading blockchain (Zhang et al., 2022).

To alleviate the problems of low consensus efficiency and network splitting in the Raft consensus algorithm, Huang et al. (2019) proposed a model of network splitting to be integrated into the Raft algorithm. They predicted the time and probability of network splitting and reduced its impact by optimizing the relevant parameters. Fu et al. (2021), aiming at the problem of performance degradation caused by the blockchain backup mechanism, proposed an improved AdRaft consensus algorithm from the aspects of log replication and leader election. The improved AdRaft improves throughput and reduces latency. Wu and Zhong. (2021) used a random election timeout method to resolve voting disagreements, avoiding the occurrence of multiple nodes obtaining the same number of votes when the number of nodes is not large. Wang et al. (2019) optimized the election of leading nodes and the consensus process in the Raft algorithm based on the Kademlia protocol and accelerated the election of leading nodes. Huang et al. (2021) improved the Raft consensus algorithm using network grouping to improve consensus efficiency in large-scale network environments. Zou et al. (2022) improved the Raft consensus mechanism based on credit scoring and accelerated the election of the Raft leader. However, the node performance, green-certificate trading, and carbon emission trading in the distributed power trading process were not comprehensively considered in these studies of the Raft consensus mechanism. In the distributed power market, the distributed energy output is difficult to predict (Zhang et al., 2023; Zong et al., 2023), making transactions prone to default. The scheduled performance of power transactions is the prerequisite for power trading to stabilize power and the overall balance of power. Green-certificate trading and trading carbon emissions encourage renewable energy development and promote the consumption of green energy.

Some enterprises have committed to using 100% green power to enhance their corporate image, obliging them to purchase green certificates to prove that they are using green power (although their power-purchasing strategies may change). At the same time, the constraints of carbon emission rights make fossil energy more expensive, affecting the power generation of fossil energy companies and the power-purchase decisions of power users. Renewable energy generation, such as wind and photovoltaic, can reduce carbon emissions (Yang et al., 2023), prompting enterprises to transition to renewable energy. Green certificates and carbon emissions trading can thus change a node's power generation and power-purchase decisions, which affects distributed power trading. These factors must therefore be considered in distributed power trading and its consensus process. It is thus urgent to combine distributed power trading with the trading of green certificates and carbon emissions.

The current consensus mechanism for distributed power trading does not comprehensively consider the impact of node default, green certificate trading, and carbon emissions on distributed power trading, based on the distributed energy power trading blockchain, this paper considers the performance default and the consensus of nodes in distributed power trading and how green-certificate trading and carbon emissions trading affect distributed power trading. One result of this analysis is the assignment of different weights to different nodes. We then design a weight-rating mechanism for nodes partaking in power trading and improve the Raft consensus mechanism by introducing node weights. The main contributions of this article can be summarized as follows:

1) This article proposes a node weight assessment mechanism in the distributed power trading process. When a node defaults, it needs to pay a certain amount of liquidated damages. When a node performs, it will receive liquidated damages paid by other defaulting nodes according to the weight ratio, incentivizing the node to actively perform.

2) This article improves the Raft consensus mechanism in the power trading process based on node weights, a greater node weight indicates a more reliable node and a greater probability of being elected leader. The realization of the node weight affects the consensus: the greater the weight, the greater the fraction of nodes in the consensus. The improved consensus mechanism will take less time than traditional Raft consensus mechanism as the consensus process progresses, and the improved consensus mechanism is more efficient.

The rest of this manuscript is structured as follows: Section 2 introduces the transaction process of distributed power trading, Section 3 introduces the constraints of distributed power trading, Section 4 introduces the consensus process and node weight assessment mechanism, Section 5 analyzes the proposed improved Raft consensus mechanism in different scenarios and compares it with traditional Raft consensus mechanisms, and Section 6 summarizes the conclusions.

2 Distributed power trading market transaction

2.1 Registration of trading node account

In the distributed power trading market, the power generation nodes and power consumption nodes participating in the transaction must register with the market before they can conduct the transaction. The account information N_i contains the node's trading address, transaction status, account balance, and weight, as expressed by

$$N_i = [A_i, T_i, B_i, W_i], \tag{1}$$

where A_i is the node's trading address, T_i is the node's transaction status, B_i is the node's transaction status, and W_i is the node's weight. The node weight is divided into two parts:

$$W_i = [W_{i1}, W_{i2}], \tag{2}$$

where W_{i1}, W_{i2} are the two parts of the weight. Node i is divided into power generation node f and power consumption node y. The node weights are updated as follows:

(1) The power generation nodes and the power consumption nodes trade in the distributed power market. When the trading terminates, the system updates the node weights based on the contracts between the two parties and the actual trading situation.

(2) After a round of consensus (i.e., a consensus iteration), the node weights are updated based on their participation in the consensus.

(3) After a certain number of consensus iterations, the partial weights of all nodes are decayed once.

2.2 Node trading process in distributed power market

(1) Publishing transaction information.

Power trading uses the day-ahead market. With a day as the time scale, power consumption nodes and power generation nodes trade freely within the trading market. According to demand, the two sides negotiate the next day's contracted power, contracted unit price, power consumption time, and other information. After the two parts of a node negotiate the details of the transaction, it applies for the transaction.

(2) Generate smart contracts.

Smart contracts are the foundation of blockchain distributed applications, providing a new and decentralized mechanism for fulfilling contracts in distributed power trading markets. They enable transactions in untrusted environments using consensus algorithms and software-controlled verification rules (Górski, 2022; Honari et al., 2023). After both parties to the transaction negotiate the transaction-related information and apply for the transaction, they generate a smart contract for the negotiated content and submit it to the blockchain for storage. At the same time, they pay a certain handling fee for the transaction.

(3) Trading and updating weights.

After generating a smart contract and the negotiated transaction time expires, both parties transmit the negotiated amount of electric power through the distribution grid. Electric power transmission stops at the negotiated transaction time. When electric power transmission ends, the smart contract settles the transaction according to the parties' actual power consumption and power generation. The smart contract then calculates the transaction amount to complete the delivery of electricity, and the power grid collects the corresponding over-the-grid fee based on the actual transaction volume, which is approved by the price authority of the region where the distributed power trading node is located. Once the transaction is completed, the blockchain updates the node weights based on the transactions of both nodes.

(4) Payment and distribution of fines.

After the transaction and power delivery and after the weight is updated, any defaulting node pays a penalty, part of which compensates for the loss of the defaulted node and the other part of which is distributed to other nodes in the chain according to the ratio of the corresponding weights. If both parties to the transaction default, the penalty paid by both parties is distributed to other nodes on the chain in proportion to their weights.

3 Distributed power trading constraints

3.1 Power-purchase balance constraints for power generation

The purchase balance constraints for power generation is

$$\sum_{f \in F} A_f^k - \sum_{c \in C} L_c^k = \sum_{y \in Y} A_y^k, \tag{3}$$

where F, C, Y are the set of power generation nodes, transmission lines, and power consumption nodes, respectively, and $\sum_{f \in F} A_f^k$, $\sum_{y \in Y} A_y^k$, $\sum_{c \in C} L_c^k$ are the total amount of electricity generated and consumed in the transaction and the line losses, respectively.

3.2 Generating unit output constraints

The unit output constraints are

$$P_g^{\min} \le P_g^k \le P_g^{\max}, \tag{4}$$

$$Q_g^{\min} \le Q_g^k \le Q_g^{\max}, \tag{5}$$

where $P_g^{\min}$ and $P_g^{\max}$ are the minimum and maximum values of the active output of the generator set g, and $Q_g^{\min}$ and $Q_g^{\max}$ are the minimum and maximum values of the reactive output of the generator set g. Both new energy generator sets and traditional energy generator sets must satisfy the active and reactive output constraints

3.3 Constraints on the number of green certificates issued

China's green certificates are issued by the National Renewable Energy Information Management Center, and each green certificate represents 1 MW h of green energy. The number of green certificates issued in t trading cycles is

$$A_{gc} = \sum_{r \in R} \sum_{k=1}^{t} \frac{A_r}{1000}, \tag{6}$$

where R is the set of renewable energy nodes applying for green certificates, and A_r is the power generated in kW h by the power generation node r.

4 Model of improved raft consensus based on node weights

4.1 Improved raft consensus based on node weights

4.1.1 Improved election of raft leader and consensus

Before the consensus starts, all nodes are given initial weights and all nodes followers. The leading node is elected as follows:

(1) If a follower node does not receive a heartbeat message from a leading node for a set period of time, it changes its identity to a candidate node, and increments its term to initiate a new election.

2) After a candidate node votes for itself, it sends a voting request to other nodes in parallel. Nodes that receive the request send a reply to the candidate node with their own weight information. Once the sum of the weights of the voting nodes exceeds half of the sum of the weights of all the nodes, the candidate node becomes the leading node.

(3) Once the leading node is determined, it sends a heartbeat message to all other nodes to establish itself as the leading node. All other nodes become followers and the election ends.

After the leading node is elected, the consensus proceeds as follows:

(1) The client sends a request to the leading node, which adds the request to the log and sends it to all followers via a heartbeat message.

(2) Each follower receives the log from the leading node and copies it locally after confirming that it is correct. The follower then sends a message to the leader for confirmation after successful copying.

(3) When the sum of the weights of the nodes from which the leading node receives an acknowledgment exceeds half of the sum of the weights of all the nodes, the leading node applies the log and returns the results of the client's execution. At this point, the nodes in the system reach a consensus on the request.

(4) The leading node waits for replies from other nodes until all follower nodes replicate the log.

4.1.2 Calculation of node timeout

In the traditional Raft consensus mechanism, the timeout of a node is randomized so that each node has approximately the same probability of being selected as leading node after a follower node has not received a heartbeat message from the leading node for some time. In the improved Raft consensus mechanism, the time of a node is calculated based on the weight of the node:

$$T_{r,i} = 100 + 5n_i, \tag{7}$$

where $T_{r,i}$ is the timeout time of node i in milliseconds, n_i is the ordering of the node weights in terms of their position among all node weights ($n_i = 1$ when the node weight is largest, $n_i = 2$ when the node weight is second largest, and so on until the last node with the smallest weight). In the improved time calculation, the node with the greatest weight has the shortest timeout and the highest probability of being selected as leading node.

4.2 Mechanism for node weight assessment

Node weight is an index used to rate the importance of nodes in a distributed power trading market. Node weight determines the fraction of a node's votes in the consensus. In the Raft consensus mechanism based on node weights, node weights are determined by a node's performance, its transaction volume, its participation in the

consensus process, the green-certificate transaction, carbon emission rights, and the failure of the transmission line connected to the generation node. The maximum node weight is 100 and the minimum is zero. The power generation node weights are calculated using

$$W_f = W_{f1} + W_{f2}, \tag{8}$$

$$W_{f1} = \alpha_1 W_{a,f} + \beta_1 W_{d,f} + \gamma_1 W_{g,f}, \tag{9}$$

$$W_{f2} = \eta_1 W_{gc,f} + \mu_1 W_{ce,f} + \lambda W_b, \tag{10}$$

and the weights of the power consumption nodes are calculated using

$$W_y = W_{y1} + W_{y2}, \tag{11}$$

$$W_{y1} = \alpha_2 W_{a,y} + \beta_2 W_{d,y} + \gamma_2 W_{g,y}, \tag{12}$$

$$W_{y2} = \eta_2 W_{gc,y} + \mu_2 W_{ce,y}, \tag{13}$$

where W_f, W_y are the total node weights of power generation nodes and power consumption nodes, respectively, $W_{a,f}$, $W_{a,y}$ are the node performance default credit weights, respectively, $W_{d,f}$, $W_{d,y}$ are the node trading volume weights, respectively, $W_{g,f}$, $W_{g,y}$ are the node participation consensus weights, respectively, $W_{gc,f}$, $W_{gc,y}$ are the node green-certificate trading calculation weights, respectively, $W_{ce,f}$, $W_{ce,y}$ are the node carbon emission weights calculation weights, respectively, and W_b is the generation node line failure calculation weight. α_1, α_2, β_1, β_2, γ_1, γ_2, η_1, η_2, μ_1, μ_2, λ are moderating factors used to adjust the proportion of each component's weight. Based on the impact of each component on the distributed power trading process, these moderating factors are used to adjust the proportion of each component's weight.

4.2.1 Node credit weighting assessment

The price of electricity in distributed power trading is negotiated between the power user and the power generator, and the agreed content generates a smart contract that is verified by the blockchain. If one party defaults on the contract, the other party suffers economic loss. When the actual power consumption of the power user is less than the negotiated power, the excess power generated is sold to the grid at a purchase price generally below the negotiated price: the power generator thus suffers economic losses. When the actual power generated by the generating party is less than the negotiated power, the power user must purchase the difference from the grid at a price that is generally above the negotiated price: the power user thus suffers economic losses. When a node performs normally, the node is rewarded with a credit, so the node's credit weight increases. When the node fails to perform normally, the node is penalized with a debit, and the node's credit weight decreases.

The formula for the credit incentive when the node performs normally is

$$W_{a,i}^k = W_{a,i}^{k-1} + \Delta R_p, \tag{14}$$

$$\Delta R_p = \left(W_a^{max} - W_{a,i}^{k-1}\right) \frac{E_{c,i}^k}{\sum E_{c,i}^k}, \tag{15}$$

where $W_{a,i}^k$ and $W_{a,i}^{k-1}$ are the credit weight of node i after transaction k, W_a^{max} is the upper weight limit, $E_{c,i}^k$ is the negotiated transaction volume of node i for transaction k, and $\sum E_{c,i}^k$ is the total negotiated transaction volume of transaction k.

The credit penalty formula in the case of node default is

$$W_{a,i}^k = W_{a,i}^{k-1} - \Delta R_b, \tag{16}$$

$$\Delta R_b = W_{a,i}^{k-1} \frac{E_{c,i}^k - E_{a,i}^k}{E_{c,i}^k}, \tag{17}$$

where $E_{a,i}^k$ is the actual transaction volume of node i for transaction k.

4.2.2 Weight rating of node transaction volume

Node transaction volume affects the node weight, which is calculated as follows:

$$W_{d,i}^k = \frac{\min\left\{E_{a,i}^k, E_{c,i}^k\right\} - \varphi \times \min\left\{E_a^{min}, E_c^{min}\right\}}{E_c^{max} - \varphi \times \min\left\{E_a^{min}, E_c^{min}\right\}} \times 100, \tag{18}$$

where E_a^{min} and E_c^{min} are the actual minimum transaction volume and the negotiated minimum transaction volume in this transaction, respectively, E_c^{max} is the negotiated maximum transaction volume in this transaction, and φ is the adjustment factor. The node transaction weight increases as the node transaction volume increases.

4.2.3 Weight rating of node participation in consensus

For the nodes to reach a consensus, the leading node sends the client's request to the follower nodes via a heartbeat message. When the leading node receives a reply from follower node i, the sum of the weights of the leading node and the replying follower nodes exceeds half of the sum of the weights of all the nodes, so the leading node applies the log. The replies from follower node i and all previous follower nodes that replied are called valid replies, and their node weights are increased as follows:

$$W_{g,i}^k = W_{g,i}^{k-1} + \omega \frac{k+1}{k}, \tag{19}$$

where ω is a moderating factor to regulate the effect of participation in reaching a consensus on the node weight. To prevent unlimited growth of this weight, the maximum of $W_{g,i}^k$ is 100. The maximum value is assigned if the limit is exceeded.

4.2.4 Weight rating of node green-certificate transaction

The green certificate, as the confirmation and proof of the attributes of non-hydro renewable energy generation and the only certificate for consuming green power, reflects the importance of the enterprise to the "double carbon" target and its concern for the interests of society. It is considered in the calculation of the node weights as follows:

$$W_{gc,i}^k = \frac{A_{gc,i}^k}{\sum_{i \in I} A_{gc,i}^k} \times 100, \tag{20}$$

where I is the set of all nodes and $A_{gc,i}^k$ is the number of green certificates of node i after transaction k. When a node sells green certificates, it has already obtained the corresponding income, and the green certificates sold are no longer involved in calculating weights. When a node purchases green certificates through consumption, these certificates are involved in calculating weights. The node's spending on purchasing green certificates as

an additional expenditure encourages the node to change its power-purchase decision, and the income from the sale of green certificates by the power generation node will also incentivize the node to expand the scale of new energy generation. Therefore, the green-certificate transaction affects the trading of distributed electricity, so its impact on node weight should be considered.

4.2.5 Weight rating of node carbon emission rights

The cycle of carbon emissions trading is usually 1 year, and enterprises whose carbon emissions exceed their quotas must purchase carbon emission allowances. Carbon emission quotas are issued to enterprises by the government, and a certain amount of carbon emission quotas are issued to enterprises based on their historical emissions and by integrating factors such as their emission reduction and the improvement of clean, low-carbon technologies. In the evaluation of carbon emission weights, the node carbon emission weights are evaluated as follows according to the relevant situation of the previous trading cycle:

$$W_{ce,i} = \frac{A_{ce,i} - A_{ce}^{\min}}{A_{ce}^{\max} - A_{ce}^{\min}} \times 100, \tag{21}$$

$$A_{ce,i} = A_{ce,i}^{q} + A_{ce,i}^{b} - A_{ce,i}^{e}, \tag{22}$$

where $A_{ce,i}$ is the carbon emission settlement amount of node i, $A_{ce}{}^{\min}$ is the minimum carbon emission settlement among all nodes, $A_{ce}{}^{\max}$ is the maximum carbon emission settlement among all nodes, $A_{ce,i}^{q}$ is the quota amount of node i in the current year, $A_{ce,i}^{b}$ is the purchase amount of carbon emission rights of node, and $A_{ce,i}^{e}$ is the actual carbon emission amount of the node. $A_{ce,i} > 0$ ($A_{ce,i} < 0$) when the quota amount of the node and the purchasing amount of the node exceed (are less than) the actual emission amount.

Enterprises that produce traditional high-carbon-emission energy power generation need to purchase carbon emission rights from the market to meet carbon emission requirements, and this cost will raise the price of electricity and narrow the price gap with renewable electricity. At the same time, the constraints on carbon emissions also prompt companies to engage in the energy transition and reduce carbon emissions. The constraints of green-certificate trading and carbon emission rights encourage enterprises to purchase renewable power and promote energy transformation and renewable energy consumption. Therefore, it is reasonable and necessary to consider how green-certificate trading and carbon emission rights affect node weights.

4.2.6 Weight rating of generation node fault

During the transmission of electric energy through a line, the line may fail due to lightning strikes, dirt flashes, and other factors. Therefore, the failure of a line connected to the power generation node should also be included in the evaluation of node weight, which is done as follows:

$$W_{b,f} = \frac{C_{t,f} - C_{b,f}}{C_{t,f}} \times 100, \tag{23}$$

where $C_{t,f}$ is the total number of historical power transmissions by generation node f and $C_{b,f}$ is the number of failures that occurred during power transmission.

4.2.7 Node weight decay

We introduce weight decay to avoid the consequences of centralization brought about by excessively high node weights. With this mechanism, the weight of a node decreases after it transmits power a certain number of times after the consensus. The node weight decay is calculated as follows:

$$W_{a,i'}^{l+1} = W_{a,i}^{l} e^{-\delta}, \tag{24}$$

$$W_{g,i'}^{l+1} = W_{g,i}^{l} e^{-\delta}, \tag{25}$$

where $W_{a,i}^{l}$, $W_{g,i}^{l}$ are the node weights of node i after consensus l, $W_{a,i'}^{l+1}$, $W_{g,i'}^{l+1}$ are the node weights of node i before transaction $l+1$, and δ is the decay coefficient. To avoid an excessive node weight brought about by the impact of the centralization without decreasing the node weight too much, the weight decays after l rounds of consensus.

4.3 Node credit weights and node penalties

4.3.1 Calculation of node transactions

During node trading, the income of the power generation node, the expenditure of the power consumption node, and the income of the grid are calculated as detailed below. When $E_{a,f}^{k} \geq E_{a,y}^{k}$, the power generation node revenue, power consumption node expenditure, and grid revenue are respectively

$$V_f = E_{a,y}^{k} U - \left(E_{a,f}^{k} - E_{a,y}^{k}\right) S, \tag{26}$$

$$V_y = E_{a,y}^{k} (U + G), \tag{27}$$

$$V_g = E_{a,y}^{k} G - \left(E_{a,f}^{k} - E_{a,y}^{k}\right) S, \tag{28}$$

when $E_{a,f}^{k} < E_{a,y}^{k}$, the power generation node revenue, power consumption node expenditure, and grid revenue are respectively

$$V_f = E_{a,f}^{k} U, \tag{29}$$

$$V_y = E_{a,f}^{k} (U + G) + \left(E_{a,y}^{k} - E_{a,f}^{k}\right) B, \tag{30}$$

$$V_g = E_{a,f}^{k} G + \left(E_{a,y}^{k} - E_{a,f}^{k}\right) B, \tag{31}$$

where V_f, V_y, V_g are the income of the power generation node, the expenditure of the power consumption node, and the income of the grid, respectively, $E_{a,f}^{k}$, $E_{a,y}^{k}$ are the actual power generation of the generation node and the actual power consumption of the consumption node, respectively, U is the negotiated tariff, S is the price of power sold by the generator to the grid, B is the price at which the power is purchased from the grid by the consumer, and G is the over-the-circuit fee to be paid by the consumption node.

4.3.2 Calculation of node penalties

The nodes on both sides of the transaction negotiate the tariff of a transaction. If one party defaults, the other party suffers a corresponding loss. When a node defaults, the node's credit weight and its tariff are both penalized. When a node defaults, the node needs to compensate for the loss of the other node and pay an additional portion of the penalty distributed to other nodes according to the credit weight of the remaining nodes. This strategy encourages other nodes to perform because all nodes on the

TABLE 1 Results of generation node transactions.

Power generation node	Transaction node	Contract unit price (yuan/kWh)	Contracted electricity (kWh)	Actual power generation (kWh)	Income (yuan)	Grid revenue (yuan)	Node credit weights before and after transaction		M_f (yuan)	A_i (yuan)
1	16	0.61	310	310	189.10	12.40	97.67	97.98	0.00	3.24
2	17	0.63	200	200	121.40	−0.80	86.47	87.64	0.00	1.11
3	18	0.64	110	110	70.40	4.40	84.71	85.44	0.00	2.83
4	19	0.63	190	190	119.70	7.60	70.52	72.95	0.00	2.41
5	20	0.61	290	290	176.90	11.60	96.24	96.71	0.00	3.20
6	21	0.62	160	160	99.20	6.40	85.58	86.58	0.00	2.86
7	22	0.63	140	140	88.20	5.60	76.82	78.22	0.00	2.59
8	23	0.61	110	110	67.10	4.40	77.83	78.88	0.00	2.61
9	24	0.63	140	140	88.20	5.60	78.88	80.16	0.00	2.65
10	25	0.62	190	190	117.80	7.60	83.74	85.08	0.00	2.81
11	26	0.67	60	60	40.20	17.40	59.55	60.61	0.00	2.01
12	27	0.65	120	120	78.00	4.80	47.47	50.20	0.00	1.66
13	28	0.64	130	130	83.20	5.20	81.57	82.61	0.00	2.73
14	29	0.66	60	60	37.00	−2.00	58.51	59.59	0.00	1.22
15	30	0.64	100	100	64.00	4.00	47.09	49.38	0.00	1.63

blockchain participate in the security of the blockchain transaction. When both nodes are in default, then neither can receive the corresponding penalty. The formula for calculating the node penalty is

$$M_f = (B - U - G)\left(C_e - E_{a,f}^k\right)\left(\varepsilon + \Delta W_{a,f}^k\right), \quad (32)$$

$$M_y = (U - S)\left(C_e - E_{a,y}^k\right)\left(\varepsilon + \Delta W_{a,y}^k\right), \quad (33)$$

where M_f is the penalty of the power generation node, M_y is the penalty of the power consumption node, C_e is the amount of electricity negotiated between the two parties, $\Delta W_{a,f}^k$ and $\Delta W_{a,y}^k$ are the reductions of the credit weights of the generation node and the consumption node, respectively, in the course of the transaction, and ε is the penalty coefficient. When one of the parties defaults on the contract between the power consumer and the power generator, $\varepsilon = 1$; when both parties default on the contract, $\varepsilon = 0$.

The penalty formula for the defaulting node depends on the change in the credit weight of the node during the given transaction. The more the node credit weight decreases, the greater is the penalty, and the penalty to be paid by the node is controlled by the node credit weight of the node.

4.3.3 Allocation of node penalties

Part of the penalty for nonperforming nodes covers the corresponding losses of the defaulted party, while the other part is distributed in proportion to the credit weights of other nodes in the power trading market to encourage the nodes to perform actively. This is expressed as

$$A_i = \frac{W_{a,i}^k}{\sum_{i=1}^{n-2} W_{a,i}^k} F_w. \quad (34)$$

where n is the number of nodes participating in the transaction process, A_i is the penalty for nonperforming nodes allocated to the remaining $n-2$ nodes according to the ratio of credit weight, $W_{a,i}^k$ is the credit weight of the node that allocates the penalty, and F_w is the fraction of the penalty paid by the defaulting node that compensates other nodes. When one of the two parties to a transaction defaults, a portion of the penalty paid by the defaulting node compensates for the loss of the node on the other side of the transaction, and the remaining portion is allocated to the other nodes according to their credit weights. When both parties to a transaction default, the penalties are allocated to other nodes according to their credit weights.

5 Simulation

5.1 Simulation of distributed power trading

We simulate the power traded over time to verify the reasonableness and effectiveness of the node credit weights designed in this paper. In distributed power trading, we use 1 day as the trading cycle; that is, trading occurs once a day, and the power generation nodes negotiate with the power consumption nodes about the power and price of the next day's transaction. The simulated power trading network contains 15 power generation

TABLE 2 Results of consumption node transactions.

Power consump-tion node	Transaction node	Contract unit price (yuan/kWh)	Contracted electricity (kWh)	Actual power consumption (kWh)	Expense (yuan)	Grid revenue (yuan)	Node credit weights before and after transaction		M_y (yuan)	A_i (yuan)
16	1	0.61	310	310	201.50	12.40	95.57	96.16	0.00	3.18
17	2	0.63	200	180	120.60	−0.80	95.60	86.04	48.58	1.09
18	3	0.64	110	110	74.80	4.40	79.30	80.29	0.00	2.66
19	4	0.63	190	190	127.30	7.60	79.82	81.48	0.00	2.70
20	5	0.61	290	290	188.50	11.60	91.87	92.89	0.00	3.07
21	6	0.62	160	160	105.60	6.40	84.99	86.03	0.00	2.85
22	7	0.63	140	140	93.80	5.60	79.95	81.16	0.00	2.69
23	8	0.61	110	110	71.50	4.40	79.83	80.79	0.00	2.67
24	9	0.63	140	140	93.80	5.60	79.84	81.06	0.00	2.68
25	10	0.62	190	190	125.40	7.60	80.55	82.15	0.00	2.72
26	11	0.67	60	80	57.60	17.40	65.05	65.96	0.00	2.19
27	12	0.65	120	120	82.80	4.80	70.37	71.91	0.00	2.38
28	13	0.64	130	130	88.40	5.20	73.27	74.78	0.00	2.47
29	14	0.66	60	50	35.00	−2.00	64.93	54.11	30.74	1.10
30	15	0.64	100	100	68.00	4.00	61.38	63.05	0.00	2.09

TABLE 3 Results of leader node election (A).

Node number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Traditional Raft	6	5	5	9	8	5	6	8	3	4	9	7	4	11	7
Improved Raft	146	24	0	0	30	0	0	0	0	0	0	0	0	0	0

TABLE 4 Results of leader node election (B).

Node number	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Traditional Raft	7	5	8	9	3	9	8	8	5	10	5	9	6	7	4
Improved of Raft	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

nodes and 15 power consumption nodes. Among them, node 17 and node 29 were set to default during the trading process, and all nodes conduct a transaction, change their credit weight, and allocate penalties to the defaulting nodes at the end of the transaction. In this transaction process, the feed-in tariff of the power generation node is 0.4 yuan/(kW h), the power consumption node purchases power from the grid at 0.75 yuan/(kW h), and the over-the-grid fee is 0.04 yuan/(kW h). Tables 1, 2 give the simulation results.

The transaction results show that, when both nodes in a transaction perform normally, the credit weight of the node increases. The higher the credit weight, the more penalties the defaulting node receives to encourage the latter to perform normally. When one of the two parties in the transaction fails to perform, its credit weight decreases, and the performing node must buy the remaining power from the grid or sell the excess power to the grid. This causes economic loss to the performing node, so the nonperforming node pays the appropriate penalties to compensate for the node on the other side of the transaction. Among them, nodes 17 and 29 are non performing nodes, so a certain penalty needs to be paid, while the other nodes are all performing nodes, so there is no need to pay a penalty.

To summarize, in the distributed power market based on node credit weight, the power node purchase price is lower than the grid price, and the power node sale price is higher than the feed-in tariff, which encourages the nodes to trade in the distributed power market and promotes the consumption of distributed renewable energy sources. The penalties for nonperforming nodes are allocated to other trading nodes according to the credit weight of the latter to reduce the losses of the trading nodes and encourage the latter nodes to perform. This strategy is fair and maintains the stability of the distributed power trading market, which is in line with the expected goal.

5.2 Simulation of node consensus

To verify the effect of the introduction of node weights on the time-consuming consensus process, the election of the leading node and the node consensus process are simulated by using 30 nodes in Section 5.1. The nodes make 200 transactions, and a consensus is reached after each transaction. During the transaction, renewable energy generation nodes obtain green certificates according to the amount of renewable energy they generate, and these green

certificates can be sold. The power consumption nodes can also prove that they have used renewable power by purchasing green certificates. The weight of the green-certificate transaction varies, and the carbon emission rights participate in the weight calculation according to the transaction result of the previous transaction cycle. $W_{a,i}$ takes the initial value of zero, $W_a{}^{max}$ takes the value of zero, $W_{g,i}$ takes the initial value of zero, $l = 30$, $\delta = 0.05$, $\alpha_1 = \alpha_2 = 0.2$, $\beta_1 = \beta_2 = 0.2$, $\eta_1 = \eta_2 = 0.1$, $\mu_1 = 0.1$, $\mu_2 = 0.2$, $\lambda = 0.1$, $\varphi = 0.5$, $\omega = 1$. The consensus process of distributed power trading takes milliseconds, in contrast with the transaction cycle of 1 day for distributed power trading. Therefore, the consensus for distributed power trading is completed very quickly.
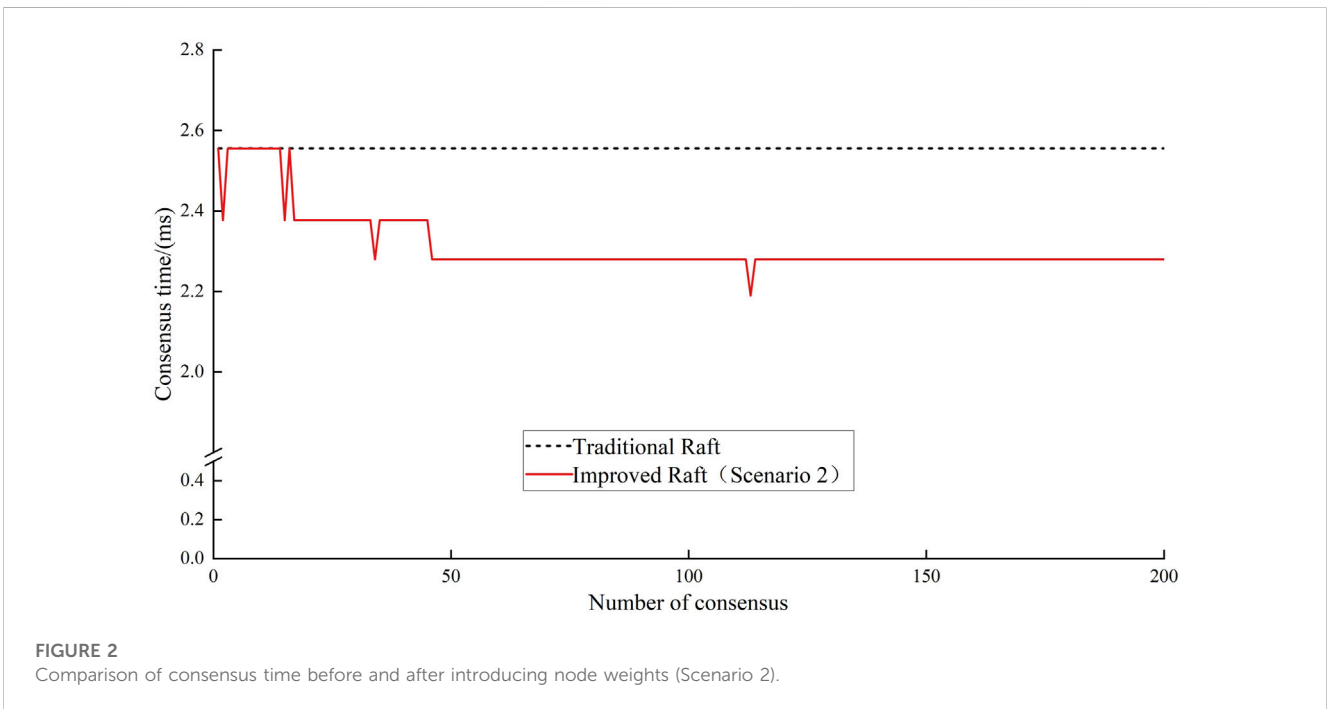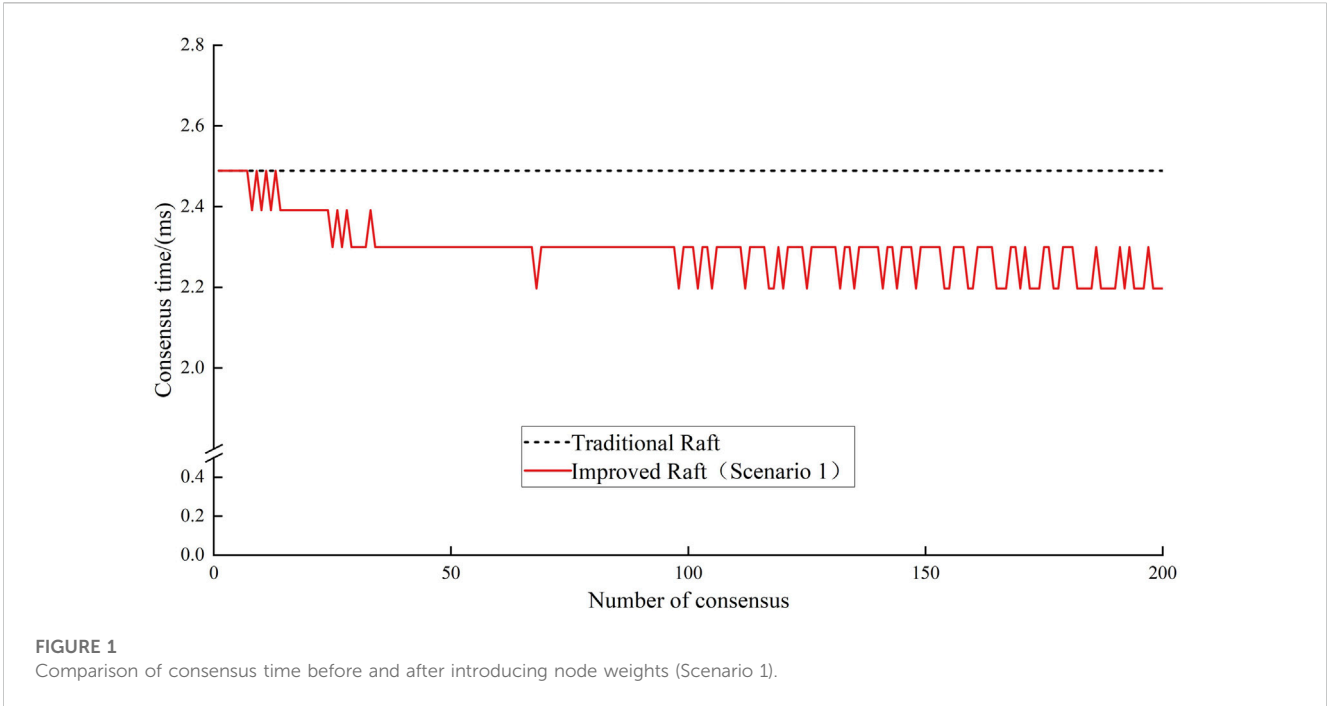
5.2.1 Comparison of election of leading node

To compare the changes in the election of the leading node before and after the improvement of Raft consensus mechanism, 200 elections were conducted using the Raft consensus mechanism before and after applying the improved election process. The number of times a node becomes a leader is then compared. To avoid the influence of node participation in consensus weight, $\gamma_1 = \gamma_2 = 0$. Tables 3, 4 show the simulation results.

The simulation results show that, during the 200 leader elections, all nodes in the traditional Raft consensus mechanism can become leading nodes, and the chances of each node becoming a leading node are relatively equal. In the improved Raft consensus mechanism, nodes 1, 2, and 5 have relatively higher weights of green certificates and carbon emission rights because they have more transactions and fewer defaults. Their total weight exceeds that of the other nodes, with 146, 24, and 30 campaigns to become the leading node, respectively. This is consistent with the design intent that nodes with high weights should have a greater probability of becoming leading nodes.

5.2.2 Comparison of time required for single node consensus

In the Raft consensus mechanism based on node weights, a heavier node has a greater share of the consensus, so we simulate the time consumed by each consensus after the introduction of node weights. We simulate the Scenario for nodes 1, 2, and 5 as leading nodes in Section 5.2.1, set one consensus after each transaction, simulate 200 transactions, and compare the times required for a single consensus. At this time, $\gamma_1 = \gamma_2 = 0.3$. Figures 1–3 show the simulation results.

FIGURE 1
Comparison of consensus time before and after introducing node weights (Scenario 1).

FIGURE 2
Comparison of consensus time before and after introducing node weights (Scenario 2).

This article sets a consensus after each transaction, so the number of consensus is equal to the number of transactions. The consensus time of the Raft consensus mechanism is determined by the node that, of all the responding nodes, has been communicating with the leading node for the longest time when the consensus is complete. This node is called the "intermediate node." With the traditional Raft consensus mechanism, the intermediate node is unchanged if the leading node does not change, so the consensus time remains unchanged each time. With the improved Raft consensus mechanism, the introduction of node weights changes the intermediate node upon changing the weights, even if the leading node does not change. Thus, the consensus time also changes. The simulation results lead to the conclusion that the improved Raft consensus mechanism based on node weights causes the consensus elapsed time to fluctuate for each succeeding consensus. However, as the consensus number increases, the role of the node consensus weights increases, so the time required for a node consensus decreases.
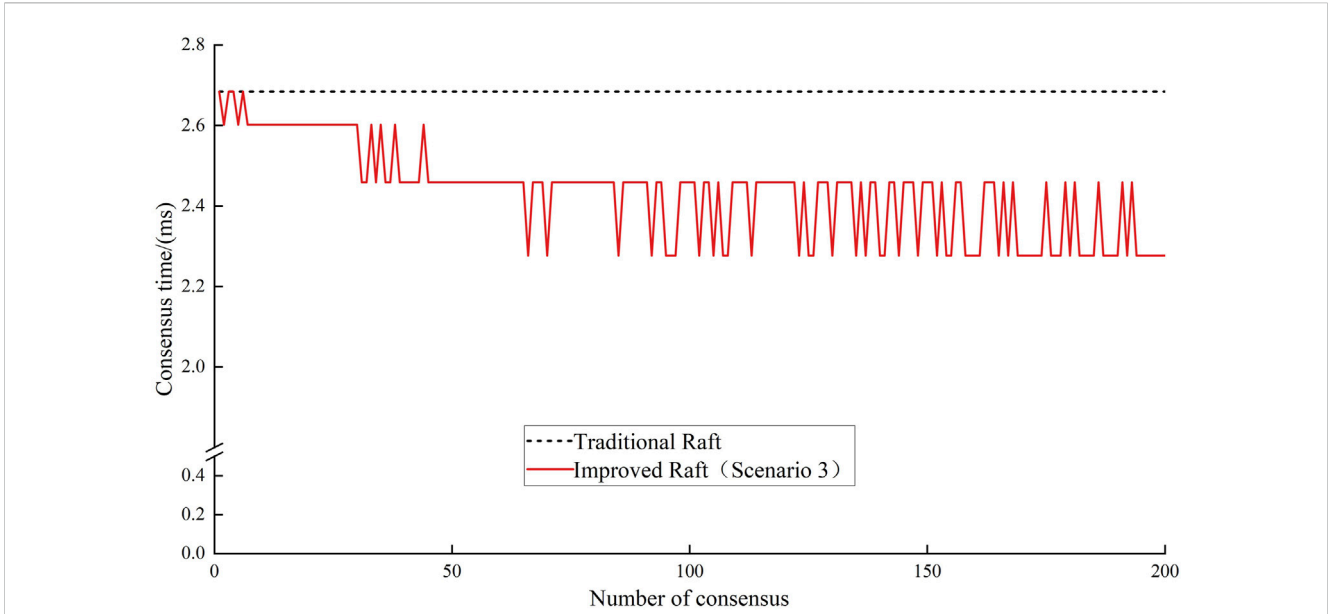
FIGURE 3
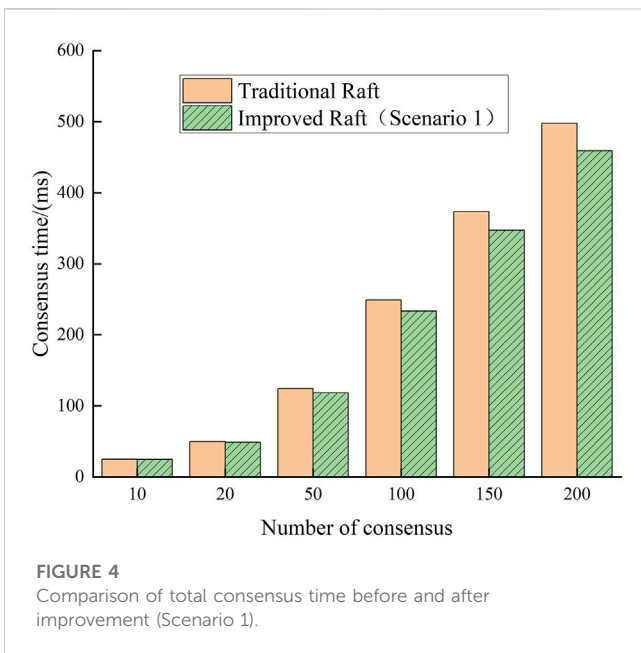Comparison of consensus time before and after introducing node weights (Scenario 3).

FIGURE 4
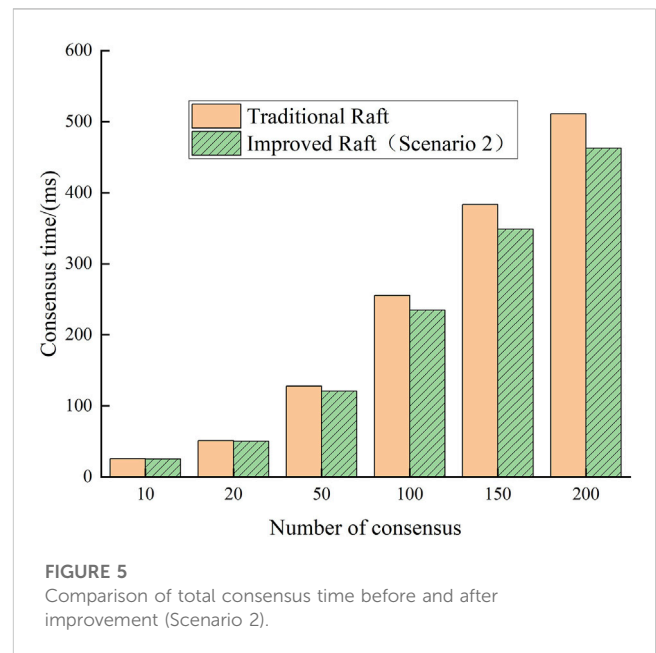Comparison of total consensus time before and after improvement (Scenario 1).

FIGURE 5
Comparison of total consensus time before and after improvement (Scenario 2).

5.2.3 Comparison of time for all node to reach consensus

To verify how node weights affect the total node consensus time, we compare (based on Sec. 5.2.2) the single node consensus time by simulating the total time for node consensus for the improved and the traditional Raft consensus mechanisms. Figures 4–6 show the simulation results. The increase in node consensus time is shown in Tables 5–7.

The simulation results lead to the conclusion that the improved Raft consensus mechanism takes less time and is more efficient than

the total consensus of the traditional Raft consensus mechanism for more consensus iterations. The efficiency improves by about 9% after 200 consensus iterations.

5.3 Changes in node weights for consensus

5.3.1 Changes in node credit weights

When trading distributed power, the credit weight of a node is an important index. To verify how node default affects the credit

FIGURE 6
Comparison of total consensus time before and after improvement (Scenario 3).
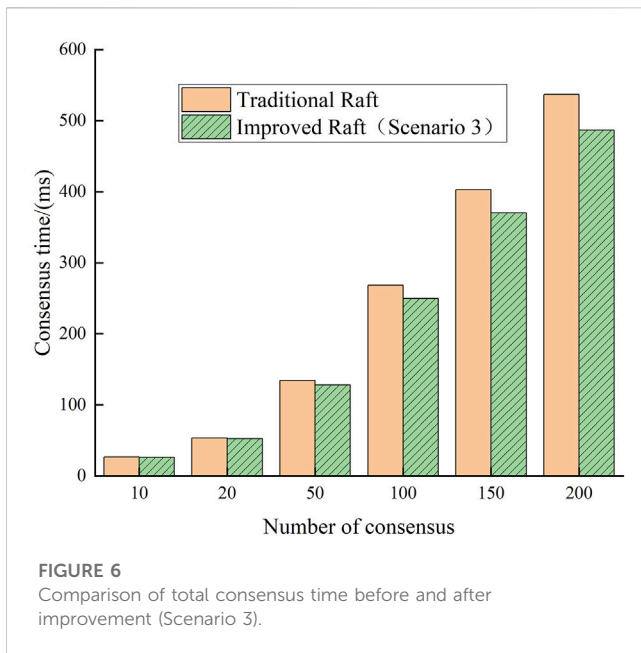
weight of a node, the change of the credit weight of the four nodes in a consensus are compared for Scenario 1 (nodes 1, 11, 17, and 27). Figure 7 shows the results.

The results show that the credit weights of the nodes are all low at the beginning and increase with increasing consensus iteration; thus, the credit weights of the nodes depend on the node's performance default. When a node has fewer defaults, the node's credit weight gradually grows at a gradually slowing rate. Fluctuations appear due to the weight attenuation, but it eventually stabilizes. When a node has many

defaults, the node credit weight decreases and fluctuates more, which is consistent with the original design intention.

For the change of node credit weight to be more intuitive, consider a set two nodes A and B. Node A has continuous compliance, and its initial credit weight is zero. Node B has continuous default and its initial credit weight is 100. Figure 8 compares the credit weight of the two nodes as a function of consensus iterations.

From the above figure, when a node continuously performs well, its credit weight gradually grows and slows, and the fluctuations decrease as the weight decays. When s node continuously defaults, the node's credit weight decreases rapidly and eventually tends to zero.

5.3.2 Changes in trading weight of nodes with green certificates

In the improved Raft consensus mechanism, nodes 1, 3, 16, and 25, which conduct green-certificate transactions, are selected under Scenario 1 to simulate variations in the weights of green-certificate transactions. Figure 9 shows the results.

Figure 9 shows that node 1 is the first node to get a green certificate. Initially, its green-certificate transaction weight rises rapidly. As other renewable energy nodes also get green certificates and green certificates begin to trade, the green-certificate transaction weight begins to fluctuate. Nodes 1 and 3 sell green certificates, and the green-certificate transaction weight rises with the number of green certificates obtained. Once a green certificate is sold, it no longer participates in the calculation of the node weight, and the weight of the green certificates declines. Nodes 16 and 25 purchase green certificates and so increase in weight. They then slowly decrease in weight as the number of green certificates in circulation varies. Nodes with more green certificates thus have greater weights, which encourages power generation nodes to produce more renewable

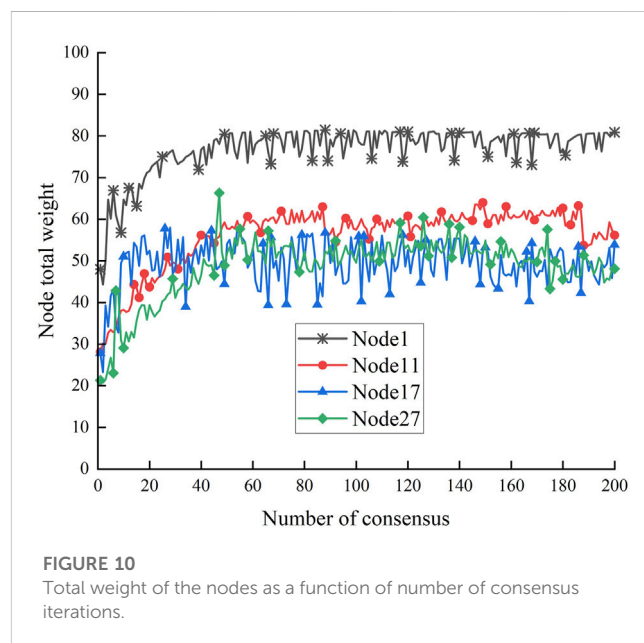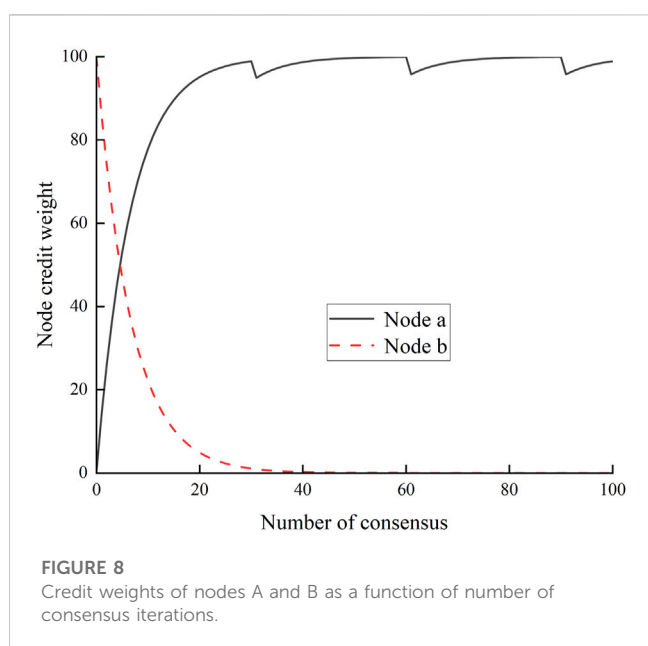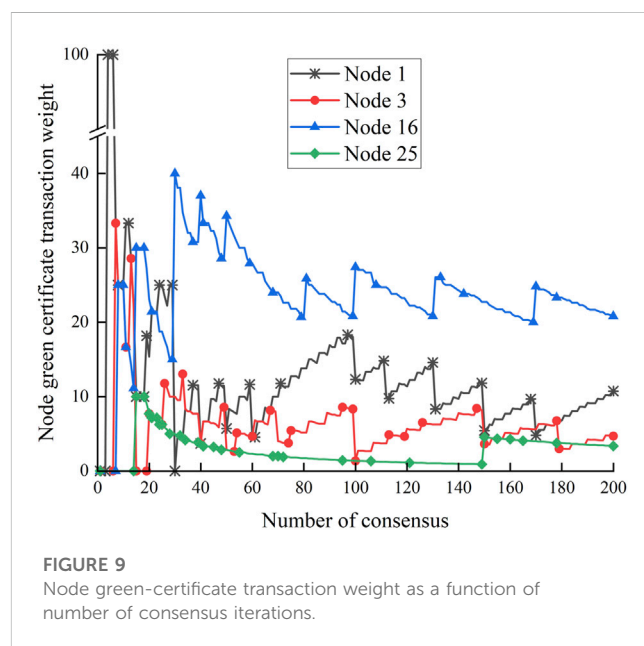**TABLE 5 Comparison of total consensus time before and after improvement (Scenario 1).**

Number of consensus	Consensus time before improvement (ms)	Consensus time after improvement (ms)	Efficiency gains (%)
10	24.89	24.70	0.78
20	49.78	48.81	1.96
50	124.45	118.45	4.83
100	248.91	233.24	6.30
150	373.36	347.00	7.06
200	497.82	459.13	7.77

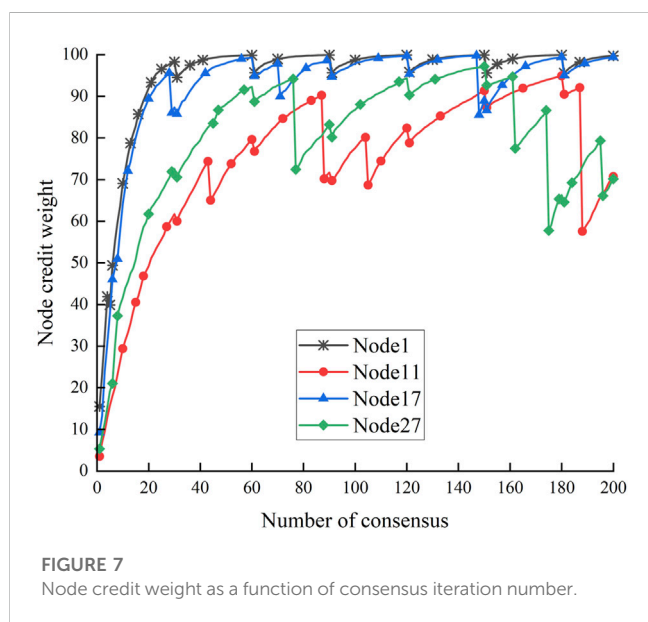**TABLE 6 Comparison of total consensus time before and after improvement (Scenario 2).**

Number of consensus	Consensus time before improvement (ms)	Consensus time after improvement (ms)	Efficiency gains (%)
10	25.56	25.38	0.70
20	51.12	50.05	2.09
50	127.79	120.79	5.48
100	255.58	234.78	8.14
150	383.37	348.67	9.05
200	511.16	462.66	9.49

TABLE 7 Comparison of total consensus time before and after improvement (Scenario 3).

Number of consensus	Consensus time before improvement (ms)	Consensus time after improvement (ms)	Efficiency gains (%)
10	26.85	26.35	1.84
20	53.70	52.38	2.46
50	134.24	128.16	4.53
100	268.48	249.86	6.94
150	402.72	370.09	8.10
200	536.96	486.67	9.36

FIGURE 7

Node credit weight as a function of consensus iteration number.

FIGURE 9

Node green-certificate transaction weight as a function of number of consensus iterations.

FIGURE 8

Credit weights of nodes A and B as a function of number of consensus iterations.

FIGURE 10

Total weight of the nodes as a function of number of consensus iterations.

power and power consumption nodes to use more renewable power, thereby promoting green consumption.

5.3.3 Changes in total node weight

With the improved Raft consensus mechanism based on node weights, the total node weight affects the consensus process of the nodes. To compare the variations in total node weight, we count the variations in the total weights of nodes 1, 11, 17, and 27 in the consensus in the case of Scenario 1. Figure 10 shows the results.

The simulation results lead to the conclusion that the total node weight gradually grows from the initial value to a threshold value and then fluctuates within a certain range, proving that the total node weight remains in a controllable range. Due to the joint influence of multiple weighting factors, the total weight of nodes will not grow without limit, which leads to over-centralization.

6 Conclusion

This paper improves the traditional Raft consensus mechanism for the transaction characteristics of distributed power trading. Nodes' green-certificate trading situation and carbon emissions trading situation affect their weight. We improve the Raft consensus mechanism for distributed power trading and simulate the market, which leads to the following conclusions:

(1) In the distributed power trading market, the power consumption nodes and the power generation nodes negotiate the tariffs and power volume of the transaction. After the transaction, the nodes that perform well are rewarded, and the nonperforming nodes are penalized by lowering their credit weight and being obliged to pay penalties. These payments are distributed to the performing nodes of the power trading market, which encourages all nodes in the chain to perform well. The system also maintains fairness in the distributed power market.

(2) To elect leading nodes, node weights are considered in combination with node transaction fulfillment, green-certificate transactions, carbon emission rights transactions, etc. Nodes with greater weight have smaller timeouts. The simulation shows that the improved Raft consensus mechanism grants nodes with greater weight and more reliable nodes a greater probability of becoming the leader. This contrasts with the traditional Raft consensus mechanism whereby each node has roughly the same probability of becoming the leader. The improved Raft consensus mechanism is thus consistent with the market goal.

(3) The consensus process of a node depends on the node weight: as the number of consensus iterations increases, the role of the node consensus weight increases. In addition, the node single consensus time decreased and improved efficiency.

(4) Comparing the nodes' green-certificate trading weights shows that increasing the number of green certificates of a node increases its weight, thus promoting the energy transition of power generation nodes and the consumption of renewable power in the distributed power trading market. The constraint of carbon emission rights also encourages nodes to comply with carbon emission constraints by purchasing carbon emission quotas.

The weight rating mechanism of nodes designed in this paper considers factors such as green certificates and carbon emission rights and motivates nodes to perform well. It also curbs the default behavior of nodes and produces safe and stable trading in the distributed power trading market. Nodes gain in the distributed power trading process and losses caused by nonperforming nodes are reduced, encouraging nodes to trade in the distributed power trading market, thereby promoting renewable energy consumption. In electing leading nodes, nodes with greater weight and reliability are more likely to become leading nodes. In addition, the consensus process among nodes accelerates the time-consuming distributed power trading consensus process and improves efficiency. The introduction of green-certificate trading and carbon emission weights motivates enterprises to consume renewable energy. It also constrains nodes to achieve carbon emission compliance by purchasing carbon emission quotas, which motivates enterprises to transform into low-carbon enterprises. However, the improved Raft consensus mechanism proposed in this article, which considers green card trading and carbon emission rights, does not consider the situation of malicious nodes attacking the distributed power trading process. The Raft consensus mechanism can ensure system consistency without malicious nodes, but this feature reduces its application scope. The improved Raft consensus mechanism in this article is also improved without malicious node attacks, without considering the situation of malicious nodes attacking distributed power trading. In the future, possible malicious node attacks in the Raft consensus mechanism of distributed power trading can be studied to improve the consensus mechanism and ensure the safe and stable operation of the distributed power trading market.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

XX: Conceptualization, Investigation, Methodology, Supervision, Validation, Writing–original draft, Writing–review and editing. YL: Investigation, Validation, Writing–original draft, Writing–review and editing. HQ: Software, Supervision, Writing–review and editing. YC: Investigation, Supervision, Writing–review and editing.

Funding

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Castro, M., and Liskov, B. (2002). Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.* 20 (4), 398–461. doi:10.1145/571637.571640

Chen, Q., Wang, K., Chen, S., and Xia, Q. (2018). Transactive energy system for distributed agents: architecture, mechanism design and key technologies. *Autom. Electr. Power Syst.* 42 (03), 1–7. doi:10.7500/aeps20171031002

Cui, Y., Zhou, H., Zhong, W., Li, H., and Zhao, Y. (2020). Low-carbon scheduling of power system with wind power considering uncertainty of both source and load sides. *Electr. Power Autom. Equip.* 40, 85–93. doi:10.16081/j.epae.202009019

Fang, X., Ma, D., Hou, W., Sun, Z., Yang, X., and Liu, J. (2019). Research on the blockchain consensus mechanism for distributed renewable energy access. *Zhejiang Electr. Power.* 38 (07), 1–6. doi:10.19585/j.zjdl.201907001

Fu, W., Wei, X., and Tong, S. (2021). An improved blockchain consensus algorithm based on raft. *Arab. J. Sci. Eng.* 46 (9), 8137–8149. doi:10.1007/s13369-021-05427-8

Górski, T. (2022). Reconfigurable smart contracts for renewable energy exchange with re-use of verification rules. *Appl. Sci.* 12 (11), 5339. doi:10.3390/app12115339

Guo, H., Gong, D., Zhang, L., Wang, F., and Du, D. (2022). Hierarchical game for low-carbon energy and transportation systems under dynamic hydrogen pricing. *IEEE Throughput Ind. Inf.* 19 (2), 2008–2018. doi:10.1109/TII.2022.3190550

Honari, K., Rouhani, S., Falak, N. E., Liu, Y., Li, Y., Liang, H., et al. (2023). Smart contract design in distributed energy systems: a systematic review. *Energies* 16 (12), 4797. doi:10.3390/en16124797

Huang, D., Li, L., Chen, B., and Wang, B. (2021). RBFT: a new Byzantine fault-tolerant consensus mechanism based on Raft cluster. *J. Comm.* 42 (03), 209–219. doi:10.11959/j.issn.1000−436x.2021043

Huang, D., Ma, X., and Zhang, S. (2019). Performance analysis of the raft consensus algorithm for private blockchains. *IEEE Trans. Syst. Man. Cybern. Syst.* 50 (1), 172–181. doi:10.1109/TSMC.2019.2895471

Huckle, S., Bhattacharya, R., White, M., and Beloff, N. (2016). Internet of things, blockchain and shared economy applications. *Procedia Comput. Sci.* 98, 461–466. doi:10.1016/j.procs.2016.09.074

Kim, J., and Dvorkin, Y. (2019). A P2P-dominant distribution system architecture. *IEEE Trans. Power Syst.* 35 (4), 2716–2725. doi:10.1109/TPWRS.2019.2961330

King, S., and Nadal, S. (2012). *Ppcoin: peer-to-peer crypto-currency with proof-of-stake.* self-published paper, August 19 1.

Lamport, L. (2019). *"The part-time parliament"* in Concurrency: the works of Leslie Lamport. New York, United States, October: IEEE.

Li, Y., Gao, D. W., Gao, W., Zhang, H., and Zhou, J. (2020). Double-mode energy management for multi-energy system via distributed dynamic event-triggered Newton-Raphson algorithm. *IEEE Throughput Smart Grid.* 11 (6), 5339–5356. doi:10.1109/TSG.2020.3005179

Li, Z., Zhao, S., and Liu, J. (2019). Coordinated optimal dispatch of wind-photovoltaic-hydro-gas-thermal-storage system based on chance-constrained goal programming. *Electr. Power Autom. Equip.* 39 (8), 214–223. doi:10.16081/j.epae.201908024

Luo, H. (2023). ULS-PBFT: an ultra-low storage overhead PBFT consensus for blockchain. *Blockchain-Res. Appl.* 100155, 100155. doi:10.1016/j.bcra.2023.100155

Meng, S., Sun, W., Han, D., Zhang, W., Zhang, W., Xiao, M., et al. (2020). Design and implementation of decentralized power transaction mechanism to spot market. *Power Syst. Prot. control.* 48 (07), 151–158. doi:10.19783/j.cnki.pspc.190934

Nakamoto, S. (2008). Bitcoin: a peer-to-peer electronic cash system. *Decentralized Bus. Rev.*

Ongaro, D., and Ousterhout, J. (2014). *"In search of an understandable consensus algorithm"* in 2014 USENIX annual technical conference (USENIX ATC 14). Philadelphia, United States: USENIX.

Snider, M., Samani, K., and Jain, T. (2018). Delegated proof of stake: features and tradeoffs. *Multicoin Cap.* 19, 1–19.

Sun, H., Li, T., Yang, C., Zhang, Y., Song, J., Lei, Z., et al. (2023). Research on carbon flow traceability system for distribution network based on blockchain and power flow calculation. *Front. Energy Res.* 11, 1118109. doi:10.3389/fenrg.2023.1118109

Tahir, M., Ismat, N., Rizvi, H. H., Zaffar, A., Nabeel Mustafa, S. M., and Khan, A. A. (2022). Implementation of a smart energy meter using blockchain and Internet of Things: a step toward energy conservation. *Front. Energy Res.* 10, 1029113. doi:10.3389/fenrg.2022.1029113

Tai, X., Sun, H., and Guo, Q. (2016). Electricity transactions and congestion management based on blockchain in energy internet. *Power Syst. Technol.* 40 (12), 3630–3638. doi:10.13335/j.1000-3673.pst.2016.12.002

Tushar, W., Saha, T. K., Yuen, C., Smith, D., and Poor, H. V. (2020). Peer-to-peer trading in electricity networks: an overview. *IEEE Trans. Smart Grid* 11 (4), 3185–3200. doi:10.1109/TSG.2020.2969657

Wang, J., Costa, L. M., and Cisse, B. M. (2016). From distribution feeder to microgrid: an insight on opportunities and challenges, 2016 IEEE International Conference on Power System Technology (POWERCON), Wollongong, NSW, Australia, November 2016 IEEE, 1–6. doi:10.1109/POWERCON.2016.7753897

Wang, R., Zhang, L., Xu, Q., and Zhou, H. (2019). K-bucket based Raft-like consensus algorithm for permissioned blockchain, 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, December 2019 IEEE, 996–999. doi:10.1109/ICPADS47876.2019.00152

Wu, Y., and Zhong, S. (2021). Research on raft consensus algorithm for blockchain. *Netinfo Secur.* 21 (06), 36–44. doi:10.3969/j.issn.1671-1122.2021.06.005

Xia, Q., Dou, W., Guo, K., Liang, G., Zuo, C., and Zhang, F. (2021). Survey on blockchain consensus protocol. *J. Softw.* 32 (2), 277–299. doi:10.13328/j.cnki.jos.006150

Xu, G., Bai, H., Xing, J., Luo, T., Xiong, N. N., Cheng, X., et al. (2022). SG-PBFT: a secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. *J. Parallel Distr. Com.* 164, 1–11. doi:10.1016/j.jpdc.2022.01.029

Yang, M., Hu, Y., Qian, H., Liu, F., Wang, X., Tong, X., et al. (2023). Optimization of day-ahead and intra-day multi-time scale scheduling for integrated power-gas energy system considering carbon emission. *Power Syst. Prot. control.* 51 (5), 96–106. doi:10.19783/j.cnki.pspc.221197

Zhang, H., Meng, Q., Wang, M., Li, J., and Zhang, Y. (2023a). Economic and low-carbon dispatch strategy of a hydrogen-containing integrated energy system considering thermal power units participating in green certificate purchase trading. *Power Syst. Prot. control.* 51 (3), 26–35. doi:10.19783/j.cnki.pspc.220873

Zhang, N., Wang, Y., Kang, C., Cheng, J., and He, D. W. (2016). Blockchain technique in the energy internet: preliminary research framework and typical applications. *Proc. CSEE* 36 (15), 4011–4022. doi:10.13334/j.0258-8013.pcsee.161311

Zhang, X., Feng, J., Chang, X., Wang, D., Ji, S., and Xie, K. (2022). Design and application of green power trading system based on blockchain Technology. *Autom. Electr. Power Syst.* 46, 1–10. doi:10.7500/aeps20210831002

Zhang, X., Lu, M., Li, H., Gao, F., Zhong, C., and Qian, X. (2023). Flexibility resource planning of a power system considering a flexible supply–demand ratio. *Front. Energy Res.* 11, 1194595. doi:10.3389/fenrg.2023.1194595

Zhang, Z., and Kang, C. (2022). Challenges and prospects for constructing the new-type power system towards a carbon neutrality future. *Proc. CSEE* 42 (08), 2806–2819. doi:10.13334/j.0258-8013.pcsee.220467

Zhao, Y., Wang, W., and Yan, S. (2023). Distributionally robust optimization scheduling of a joint wind-solar-storage system considering step-type carbon trading. *Power Syst. Prot. control.* 51 (6), 127–136. doi:10.19783/j.cnki.pspc.220771

Zong, X., Zou, S., Zhou, H., and Dou, X. (2023). Robust stochastic low-carbon optimal dispatch of park-integrated energy system with multiple uncertainties from source and load. *Front. Energy Res.* 11, 1226768. doi:10.3389/fenrg.2023.1226768

Zou, X., Shen, L., Zhang, W., Cao, W., and Yang, M. (2022). Improvement of RAFT consensus mechanism for power transaction blockchain based on credit scoring. *South. Power Syst. Technol.* 16 (06), 132–139. doi:10.13648/j.cnki.issn1674-0629.2022.06.014