Soft Computing (2020) 24:16593–16604
https://doi.org/10.1007/s00500-020-04962-0

METHODOLOGIES AND APPLICATION

Interaction design research based on large data rule mining and blockchain communication technology

Jiboning Zhang[1]

Published online: 28 April 2020

Abstract
As the amount of information in the internet of things increases, data storage management tends to be distributed, which leads to problems such as difficult data cooperation and interaction between sites, low communication efficiency, and poor reliability. Blockchain is one of the new information technologies supporting the development of management information system. It provides a solution for the storage, verification, transmission, and exchange of distributed data. To optimize the communication performance from two aspects of communication topology and communication mechanism, a multi-link and concurrent communication tree model was constructed. To improve blockchain communication technology, an interactive design method based on big data rule mining and blockchain communication technology was proposed, which mainly solved the optimization of transmission performance of blockchain data. On the basis of ensuring the stability and reliability of data transmission, the efficiency of data transmission in blockchain was further optimized, and the integrated factor communication tree algorithm (IFT) was proposed. To solve the influence of transmission delay between nodes on communication performance, a multi-link multifactor weighted communication tree algorithm (MMWT) considering weight was proposed. Moreover, to improve the efficiency of data communication, ensure the reliability of transmission, and improve the fairness of service in the blockchain, different strategies for optimizing the performance of data communication in the blockchain were proposed under the constraints of factors such as node communication ability, node trust, weight, and service request priority. Finally, the simulation results showed that MMWT algorithm had good communication performance in concurrent communication time, communication tree reliability, communication tree depth, and other aspects.

Keywords Data sharing · Rule mining · Blockchain · Communication technology · Interaction design · Data transmission

1 Introduction

With the continuous development of economic globalization, enterprises are developing toward transregional and transnational, and the distributed management of enterprise data has become an inevitable trend (Mengelkamp et al. 2018). The enterprises' data quantity is increasing day by day, and traditional centralized data management will lead to a heavy load on the central node, which will easily cause the failure of the central node and the phenomenon of paralysis of the whole network. More and more enterprises tend to have a distributed storage of data in various sites according to actual business requirements (Huckle et al. 2016). However, distributed data storage brings a difficult data cooperation and interaction between sites, low communication efficiency, poor reliability, and other problems, which seriously affect the enterprises' response speed and timeliness for data acquisition (Wang et al. 2017).

Enterprises are seeking for a more effective way to solve the problem faced by distributed data management (Singh et al. 2019). Blockchain is one of the emerging information technologies supporting the development of management information system, and it provides a solution for the storage, verification, transmission, and communication of distributed data. In the past 2 years, enterprises have gradually realized the broad application prospect of blockchain technology in distributed data management and

Communicated by V. Loia.

✉ Jiboning Zhang
jbnzhang14@fudan.edu.cn

[1] School of Computer Science and Technology, Fudan University, Shanghai 200000, China

excavated the huge potential of blockchain in various industries. Currently, there is no widely accepted definition of blockchain (Hussein et al. 2018). In distributed storage environment, users pay more and more attention to the timeliness of interaction experience and the reliability of information. However, the efficiency of users is often limited by the low efficiency of data communication among distributed sites. One of the important goals of distributed data management is to improve the efficiency of data transmission and ensure the reliability of data transmission. It is one of the key issues in blockchain research field that which communication model and which communication algorithm can effectively organize nodes to conduct transaction verification, thereby shortening the confirmation time of blockchain transaction and improving the processing capacity of blockchain transaction business (Meeuw et al. 2019; Puthal et al. 2018; Chung and Cha 2018).

The computing efficiency and storage space of blockchain will be improved with the continuous updating of computer hardware configuration, whereas the improvement of data transmission efficiency requires a reasonable communication model and effective data communication algorithm. To optimize the communication performance from the aspects of communication topology and communication mechanism, this paper first constructed a multi-connection concurrent communication tree model. To improve blockchain communication technology, an interactive design research method based on big data rule mining and blockchain communication technology was proposed in this paper, in order to mainly solve the optimization problem of blockchain data transmission performance. Meanwhile, blockchain data transmission paths were planned by scientific and reasonable methods, so that the data can be transmitted rapidly, reliably, and fairly in the face of the distributed data communication, thus accelerating the users' response speed, enhancing their working efficiency, and improving their experience.

2 Related work

The research on blockchain mainly starts from bitcoin, and bitcoin system is the first typical application example of blockchain technology. Although in 2016, the founders of bitcoin announced that the project was a failure, in recent years, blockchain technology, as the underlying architecture of bitcoin, has been sought after by the government and the capital market. In 2015, blockchain became the highest financing sector in American venture capital. Super ledger is an open-source project launched by Linux foundation at the end of 2015 to promote blockchain transaction verification (Cocco et al. 2017). In April 2017, Samsung Electronics released a B2B (business to business) digital commercial blockchain platform, which can provide large-scale real-time transactions, detect control information, and conditionally isolate block data to ensure data security.

Blockchain is a technology rising with the encryption of digital currency. Its main feature is to form a decentralized architecture that does not rely on a third party, thus achieving the unification of technical standards and meeting the needs of various social scenarios for data storage (Chang et al. 2019; Grover et al. 2019; Pustišek and Kos 2018). Blockchain can be regarded as a new technological revolution in the context of big data and the internet. It is committed to building a credit system-based participatory society, and sharing economy will be its best application scenario. Consensus mechanism is a key point of blockchain, which realizes trust through code design, that is, to determine the block constructor and to maintain data consistency throughout the network, which is an indispensable part of blockchain (Rahman et al. 2019). Different consensus mechanisms can be selected according to different basic chains, implemented applications or business models of blockchain.

Blockchain technology was first written in the 13th five-year plan for national informatization issued by the State Council in December 2016. In May 2017, under the guidance of the Ministry of Industry and Information Technology of China, the China blockchain technology and industry development forum published The Reference Architecture of Blockchain and Distributed Ledger Technology, which is the first domestic blockchain standard under the guidance of the government. 2017 can be regarded as the first year of blockchain, and the application value of blockchain technology in various industries is being studied all over the world. Regarding the application layer of blockchain technology, it undoubtedly originates from the financial industry, but it has been rapidly popularized in social management, efficiency improvement of energy technology, and other aspects (Xie et al. 2019). This mainly considers that centralization causes efficiency problems to the development of these fields, while blockchain can perfectly realize industry decentralization management.

In general, blockchain builds a value transmission network. In order to solve the problem of communication efficiency and reliability in the transmission layer of blockchain, the optimization of data communication performance of blockchain is discussed, hoping to effectively organize blockchain nodes for transaction verification.

3 Structural model description

3.1 Communication structure and advantages

Blockchain adopts linear or star structure for its data verification, that is, accounting node v_0 distributes blockchain data to other nodes in the blockchain for verification, and communication may result in paralysis due to the failure of communication source nodes. The algorithm studied in this paper adopts tree communication structure, which is more suitable for blockchain data forwarding (Zhao et al. 2019). For the verification of a blockchain transaction, all nodes communicate according to the self-organized tree structure of the communication tree algorithm. The tree structure does not change the network structure of realizing physical network, but it is a virtual logical structure abstracted from realizing physical network. Figure 1 shows the physical network topology of blockchain, which is a network structure. All nodes in the figure are distributed in geographically dispersed areas and connected together by wired or wireless means. The lunk in the figure represents the actual wired or wireless connection (Sikorski et al. 2017). As shown in Fig. 1, the physical topology of blockchain is a network structure, and Fig. 1 is a multi-connection concurrent communication tree, in which all nodes of the blockchain are abstracted into a tree structure for blockchain transaction verification. Therefore, it is a virtual logical structure and shields the underlying physical topology. The communication source node is responsible for building communication tree, so it is a temporary self-organizing network. All the nodes participating in the verification commonly maintain the tree structure, and each node has the function of routing and storing and forwarding data (Grover et al. 2019; Muthanna et al. 2019). The wires in Fig. 1 are a logical connection, and the wires between any two nodes ignore network relay devices such as router, wireless access device (AP), and switch.

3.2 Concurrent communication tree model

Blockchain adopts P2P network, and the physical topology of the actual network is reticulate (Cho 2018). The tree structure studied in this paper does not change the physical connection status of the network, but only organizes the nodes into a tree structure from the logical level to conduct blockchain transaction verification. The location of nodes in the communication tree and the selection of communication parent–child relationship, edge and weight all depend on the communication tree algorithm adopted by them. The forms of communication trees constructed by different communication tree algorithms are different. If the blockchain communication is completed according to different tree structural nodes, they will show different performance in terms of reliability index, efficiency index, and fairness index. Therefore, multiple performance evaluation indexes shall be adopted to evaluate communication algorithms.

In a blockchain, nodes are not only resources or service providers but also resources or service consumers, and each node has an equal status (Yang et al. 2018). After introducing concurrent communication mechanism in blockchain communication, all nodes that obtain data can undertake corresponding communication tasks. The basic idea of concurrent communication mechanism is to assign

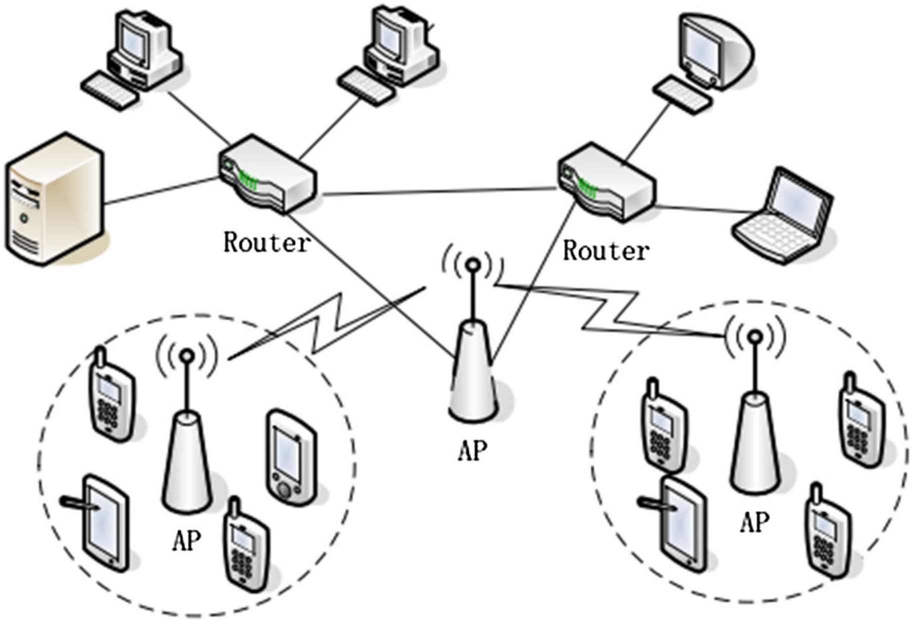


Fig. 1 Physical network topology of blockchain

appropriate communication tasks to nodes except communication source nodes with communication capability, thereby improving the communication efficiency through the concurrency of communication task execution. With the constant expansion of communication scale, the communication concurrency degree is increased significantly and communication efficiency is improved greatly (Goranović et al. 2017; Qian et al. 2018; Dubey et al. 2020). If only concurrent transmission is considered, it is a single-connection concurrent communication tree model, but after considering the heterogeneity of node communication ability, the model becomes a multi-connection concurrent communication tree model, as shown in Fig. 2.

To describe the difference between node forwarding capabilities, the concept of node communication connection number is introduced. Since any node can simultaneously forward data to other multiple nodes, the single-connection concurrent communication tree model is evolved into the multi-connection concurrent communication tree model (Kim 2018). The nodes in Fig. 1a become the node cluster in Fig. 1b. The number of nodes in any node cluster is equal to the sum of all nodes in the parent node cluster; for example, the number of nodes in cluster 1 is $l(v_0)$. The node cluster in the lower communication tree contains a great number of nodes. In order to give play to the communication ability of nodes to the maximum extent, it is required that each node v_i is in full connection communication, namely the number of communication nodes is equal to the maximum number of communication connections of the node. In addition, each node v_i forwards the data immediately after receiving the data, and there is no interval in the whole communication process.

4 Parallel clustering based on MapReduce

4.1 System structure of MapReduce

Google proposed MapReduce computing model for the first time in 2004. It helps Google rebuild the index file system, which is one of the three core technologies of Google (the other two are GFS and BigTable, respectively) (Saberi et al. 2019). MapReduce programming model allows developers to focus on data operation, and encapsulate the details of data distribution, parallelization implementation, fault-tolerant mechanism, and load balancing in a library, thus making it easier to process large data. Now, MapReduce has been widely used in the sorting of massive data, log analysis, looking for specific patterns in massive data, and other scenarios.

In Hadoop, two types of servers are mainly used to execute MapReduce task: one is JobTracker, also known as job server, and the other is TaskTracker, also known as task server (Hussen 2018). JobTracker is mainly responsible for managing and scheduling all jobs under this architecture, while TaskTracker is mainly used for executing jobs. Only one JobTracker is allowed in a Hadoop cluster, but more than one TaskTracker.

In actual operation, each MapReduce task will be instantiated as a job, and each job is divided into two phases: Map phase and Reduce phase. These two phases

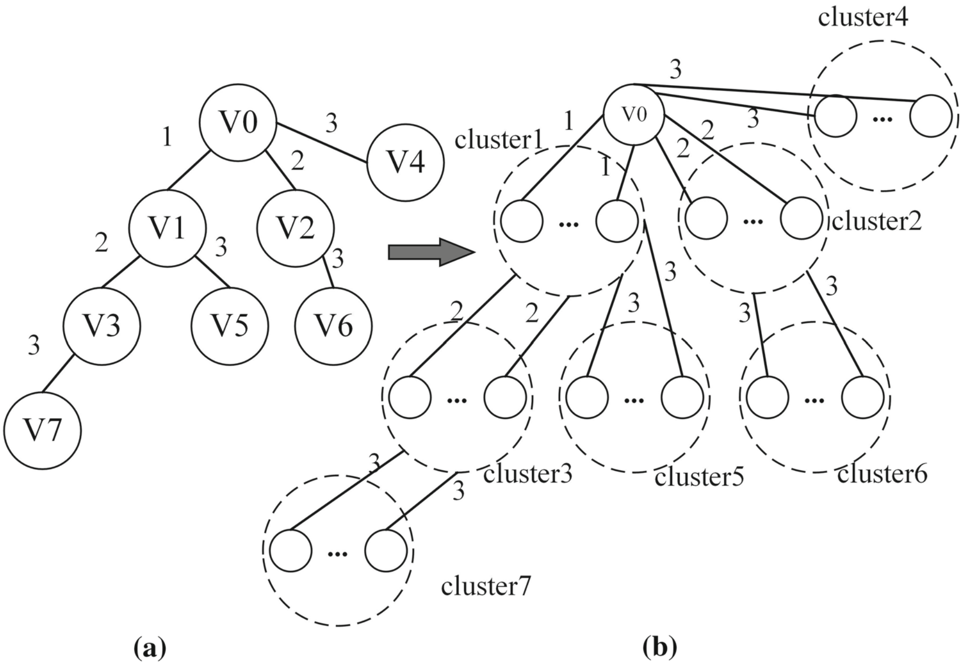


Fig. 2 Single-connection concurrent communication tree model evolves into multi-connection concurrent communication tree model

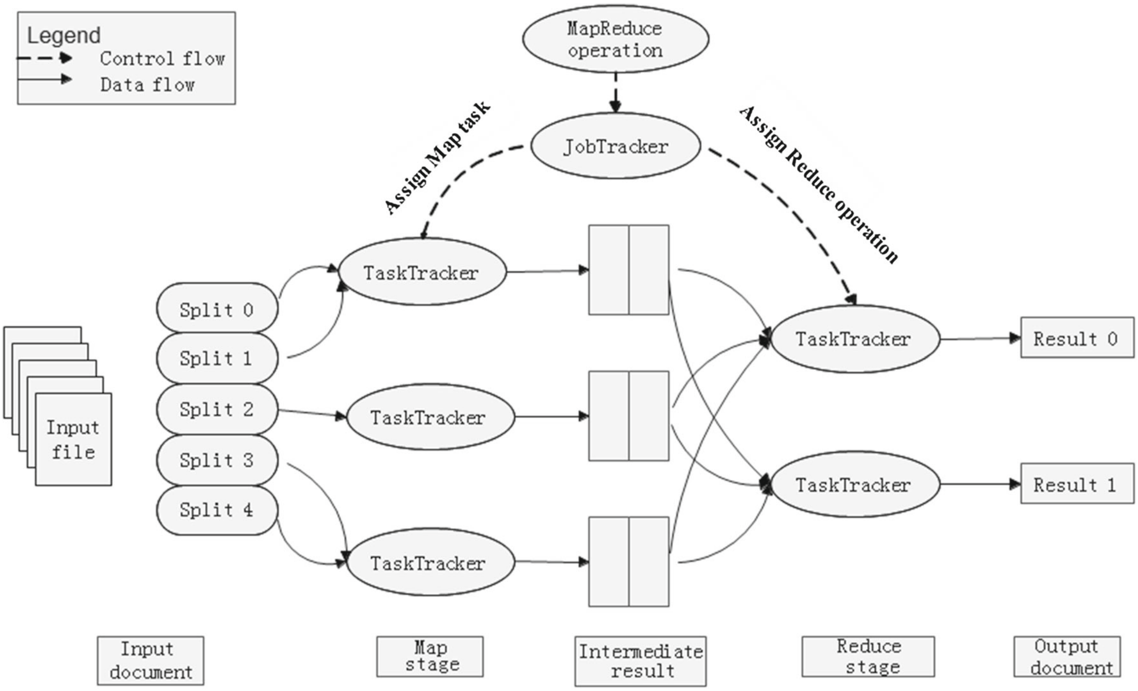


Fig. 3 MapReduce framework

are represented by Map function and Reduce function, respectively, and developers define tasks by implementing these two functions. The Map function takes the key-value pair <key, value> as input and produces an intermediate output of the set of <key, value>. Then, MapReduce library will gather all intermediate values with the same intermediate key values together and pass them to the Reduce function, which receives an input whose form is <key, (list of values)>. After processing, the final results will be output. The output form of Reduce function is still <key, value>. The system structure of MapReduce in Hadoop is shown in Fig. 3.

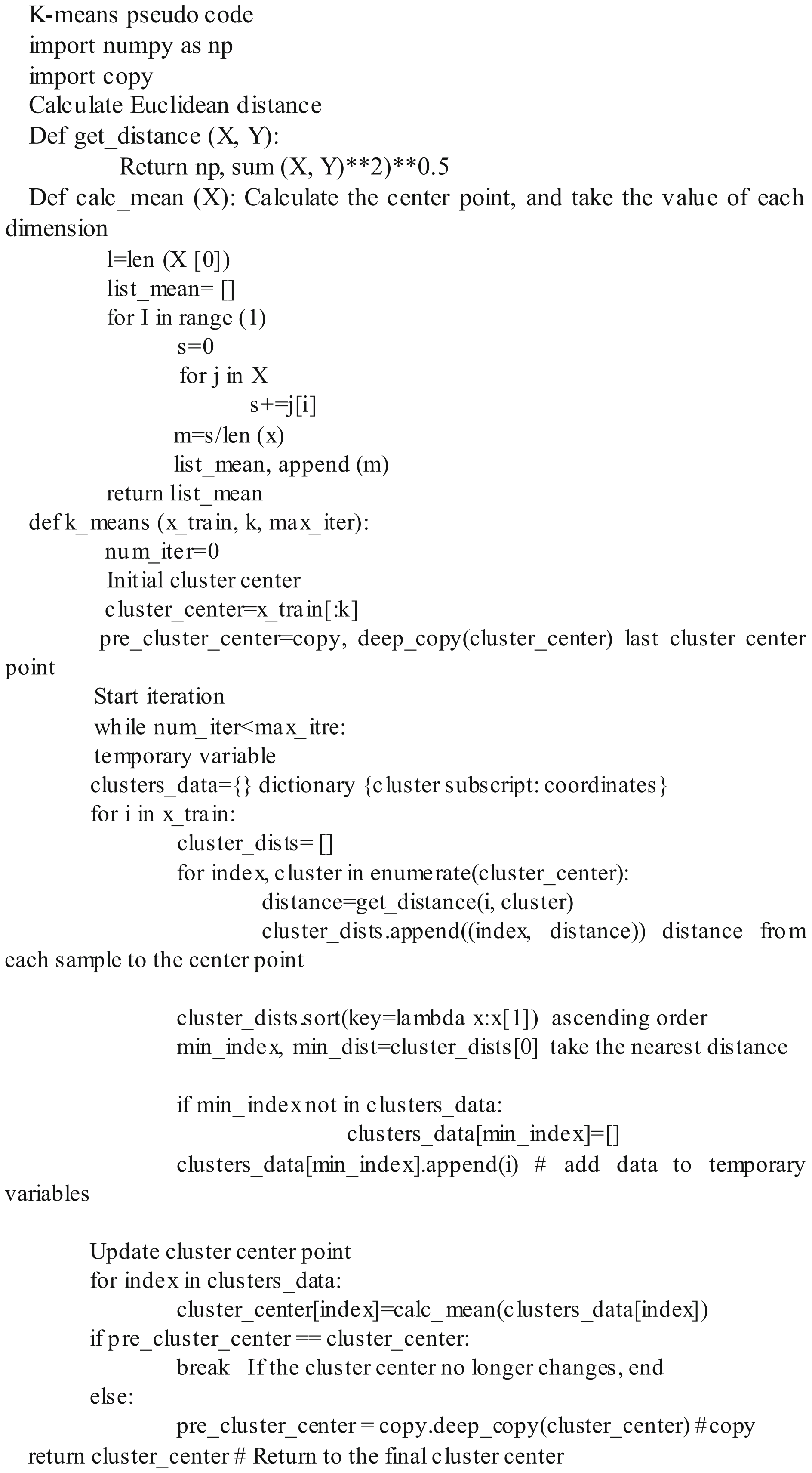
4.2 Parallel *K*-means algorithm

K-means algorithm has been proved to be able to carry out parallel processing of large data sets on Hadoop platform. MapReduce computation model is used to redesign the improved L + Cop-K-means algorithm in chapter 3, in order to design and implement Map function, Reduce function, and Combine function which is responsible for processing intermediate results. In this paper, only the allocation rules of data samples in the Map function need to be modified, and the processing of Combine function and Reduce function is the same as that of parallel K-means algorithm.

First, *K* samples are randomly selected from the data set as the central points and stored in HDFS together with prior knowledge as global variables to guide the whole clustering process. The iterative process is mainly completed by Map function, Combine function, and Reduce function.

K-means algorithm

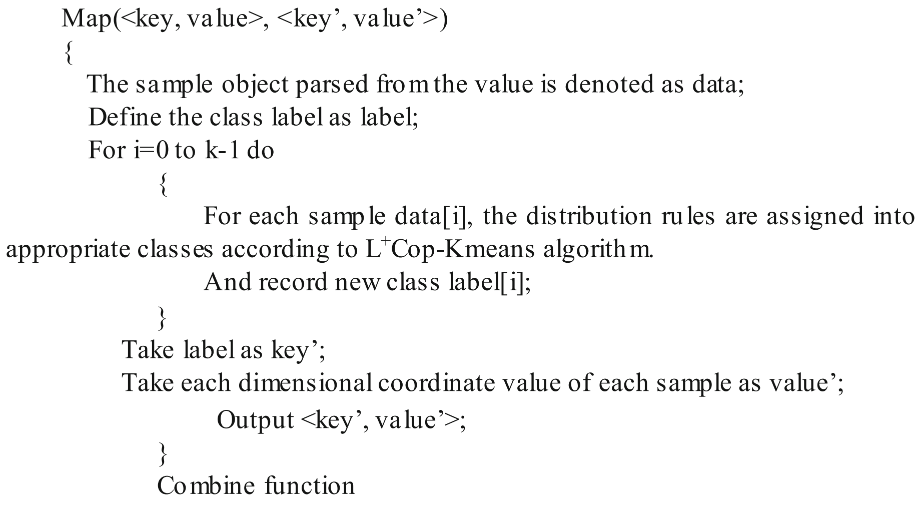
Initialize *k* cluster centers; update the cluster ownership of all sample points: the cluster to which the sample points belong most recently; recalculate the center of each cluster (until the cluster center no longer changes or reaches the maximum number of updates).



```
  K-means pseudo code
  import numpy as np
  import copy
  Calculate Euclidean distance
  Def get_distance (X, Y):
          Return np, sum (X, Y)**2)**0.5
  Def calc_mean (X): Calculate the center point, and take the value of each
dimension
          l=len (X [0])
          list_mean= []
          for I in range (1)
                s=0
                for j in X
                      s+=j[i]
                m=s/len (x)
                list_mean, append (m)
          return list_mean
  def k_means (x_train, k, max_iter):
          num_iter=0
          Initial cluster center
          cluster_center=x_train[:k]
          pre_cluster_center=copy, deep_copy(cluster_center) last cluster center
point
         Start iteration
         while num_iter<max_itre:
         temporary variable
         clusters_data={} dictionary {cluster subscript: coordinates}
         for i in x_train:
                cluster_dists= []
                for index, cluster in enumerate(cluster_center):
                        distance=get_distance(i, cluster)
                        cluster_dists.append((index, distance)) distance from
each sample to the center point

                cluster_dists.sort(key=lambda x:x[1]) ascending order
                min_index, min_dist=cluster_dists[0] take the nearest distance

                if min_index not in clusters_data:
                               clusters_data[min_index]=[]
                clusters_data[min_index].append(i) # add data to temporary
variables

         Update cluster center point
         for index in clusters_data:
                cluster_center[index]=calc_mean(clusters_data[index])
         if pre_cluster_center == cluster_center:
                break   If the cluster center no longer changes, end
         else:
                pre_cluster_center = copy.deep_copy(cluster_center) #copy
  return cluster_center # Return to the final cluster center
```

Map function

The input form of Map function is <key, value> key-value pair. Key represents the offset of the current sample data in the entire input file, and value represents the string values composed of the dimensional coordinates of the current sample. First of all, the sample values are parsed from the value, and all samples are divided into appropriate classes according to the allocation rules of stand-alone L + Cop-K-means algorithm. Then <key', value'> key-value pair is used as the output of Map function. Here, key' represents the class label after dividing each sample, and value' still represents the string value of this sample. The pseudocode of Map function is shown below:



```
Map(<key, value>, <key', value'>)
{
  The sample object parsed from the value is denoted as data;
  Define the class label as label;
  For i=0 to k-1 do
        {
                For each sample data[i], the distribution rules are assigned into
appropriate classes according to $L^+$Cop-Kmeans algorithm.
                And record new class label[i];
        }
     Take label as key';
     Take each dimensional coordinate value of each sample as value';
                 Output <key', value'>;
        }
        Combine function
```

To reduce the amount of communication (mainly data transfer between various nodes) in the iteration process, a Combine operation is required after Map function processing. The output of Map function is combined locally. Since the output of Map function is always stored locally first, the communication cost of executing Combine operation is very low.

4.3 Blockchain communication performance optimization based on IFT algorithm and MMWT algorithm

Link first communication tree (LFT) algorithm makes full use of nodes with strong communication ability to forward communication tasks and has high communication efficiency. $f(t)$ can reach the optimal solution, but does not consider the trust degree of nodes. If the nodes are added to the communication tree according to the trust degree, the algorithm is called trust first communication tree algorithm (TFT). Considering the security and stability of the whole communication tree, the TFT algorithm improves the trust of the communication tree by improving the trust of the forwarding nodes, and preferentially selects the nodes with high trust to join the top or upper layer of the communication tree. However, the nodes with high trust may have poor communication ability (Khezr et al. 2019). Thus, TFT algorithm can improve the stability of communication tree and shorten the concurrent communication time, which may lead to low communication efficiency.

MMWT algorithm takes a comprehensive trade-off between the number of communication connections, the degree of trust, and the weight of communication. It selects the target node and the communication path from the blockchain network to build the communication tree based on the accounting node. When constructing the communication tree, to greatly improve the communication concurrency, the MMWT algorithm first adds the nodes with large number of communication connections to the communication tree and then eliminates the nodes with too large communication weight in the top and upper communication trees, thus reducing the communication time (Zeng et al. 2020).

LFT: Based on the multi-connection concurrent communication tree model, with the expansion of the scale, the logical connection between the nodes in the communication tree will become complex, unable to define the communication relationship between each node. Therefore, a communication tree representation method is proposed, which transforms the communication tree into a binary tree representation. The time from the root node to the node v_i for obtaining the communication data is called the time when the node v_i obtains the data, recorded as $f_{v_i}(t)$. The cumulative communication time of a communication tree with n nodes can be expressed as $f_A(t)$:

$$f_A(t) = \sum_{i=1}^{N} f_{v_i}(t) \tag{1}$$

For the communication tree whose communication node is N, the ratio of the accumulated communication time $f_A(t)$ to n is called the average end-to-end delay of the node, recorded as $f_{\mathrm{AED}}(t)$, which can be expressed as:

$$f_{\mathrm{AED}}(t) = \frac{\sum_{i=1}^{N} f_{v_i}(t)}{N} \tag{2}$$

From the communication root node v_0 to a certain leaf node v_i receiving data, the edge passed is called the communication branch of the leaf node v_i, which is expressed as $\mathrm{Route}(v_i)$. The number of times each branch forwards the same group of data is called the link pressure of the communication branch, which is expressed as $\mathrm{Num}[\mathrm{Route}(v_i)]$. The average link pressure A_{ls} of the communication tree can be expressed as:

$$A_{\mathrm{ls}} = \frac{\sum_{i=1}^{R_n} \mathrm{Num}[\mathrm{Route}(v_i)]}{R_n} \tag{3}$$

R_n indicates the number of communication branches needed to complete the whole communication.

LFT algorithm makes full use of nodes with strong communication ability to forward communication tasks, which has high communication efficiency and $f(t)$ can reach the optimal solution. On this basis, according to the trust degree of the nodes, the nodes are added to the communication tree in turn, that is, TFT, which can improve the trust degree of the forwarding nodes and the communication tree.

MMWT: according to the actual needs, the blockchain network can be abstracted as a graph $G = (V, E, W, L, T)$, where V is the node set, E is the edge set, W is the weight set, L is the communication connection number set, and T is the node trust set. In blockchain network, if the weights of any two nodes can be obtained, G is a complete graph with weights. However, when the blockchain network scale N is large, it is difficult to calculate the transmission delay of any two nodes, and the calculation distortion is caused by the poor timeliness. For the blockchain network with N nodes, the final number of communication tree nodes is N, and the number of sides is $N - 1$, so there is more choice space for communication weight than the number of communication connections. In MMWT algorithm, nodes with large number of communication connections $l(v_i)$ are added to the communication tree first, and then, nodes with too large communication weight in the top layer and upper layer communication tree are eliminated, thus shortening the communication time.

5 Experimental analysis

5.1 Simulation experiment environment

Block data in blockchain consist of block head and block body. Block head includes version number, parent block hash value, Merkle tree root, timestamp, difficulty value, Nonce, and other information. The block body records the number of transactions in the block, the verified historical transactions, and the transactions generated in the block creation process. According to the transaction verification process of blockchain, the following agreements are made:

1. The node has the function of "staging-forwarding." Compared with the forwarding delay, the staging delay can be ignored. The transmission delay of blockchain nodes is mainly caused by forwarding delay.
2. For a transaction, the generated hash value is the same as the block data, that is, all nodes participating in the verification receive the same hash value or block data. Without considering the routing control information of nodes, it can be agreed that the task of communication subject between any two nodes in the blockchain network is the same.
3. Due to the frequent verification transactions in the blockchain, each transaction needs to be verified by all nodes, so each node knows the information of the node itself and the information between nodes in the blockchain.
4. The node obtaining the bookkeeping right needs to send all information to other nodes for verification. The transmission mode is point to multipoint. For the N-node blockchain network, the number of accounting right nodes obtained through computing power competition is 1, and the block data should be sent to other $N - 1$ nodes for verification.
5. For a transaction verification task of blockchain, all nodes do not participate in other tasks and only complete the blockchain verification task. Hence, the number of communication links of nodes is constant throughout the communication period, and the node load and available bandwidth is not considered.
6. According to whether to undertake communication task in communication, it can be divided into communication root node, forwarding node, and leaf node. The accounting authority node needs to forward the block data to other nodes, so the accounting authority node is the communication root node.

5.2 Comparison of three communication tree representation algorithms

The trust degree of the communication tree reflects the reliability and stability of the whole communication process. The greater the trust degree of the communication tree is, the higher the reliability of the communication forwarding information is, and the higher the forwarding stability is. The smaller the trust degree of the communication tree is, the worse the credibility of the forwarding information is, and the worse the link stability is. Table 1 shows the comprehensive comparison of communication performance evaluation indexes of LFT, TFT, and IFT. The IFT algorithm proposed considers the trust degree of nodes and the number of communication connections of nodes. On the premise of ensuring the reliability and stability of the communication service, it can make the concurrent communication time of the communication tree close to or reach the optimal value.

For the communication tree with the same number of nodes, the concurrent communication time of the IFT algorithm is slightly higher than that of the LFT algorithm, while that of the TFT algorithm is larger, as shown in Fig. 4. The main reason is that the TFT algorithm preferentially selects the nodes with high trust to join the communication tree, and cannot undertake the forwarding task in the subsequent communication process, so the concurrent communication time is large.

The trust degree of the communication tree reflects the reliability and stability of the whole communication process. The greater the trust degree of the communication tree is, the higher the reliability of the communication forwarding information is, and the higher the forwarding stability is. The smaller the trust degree of the communication tree is, the worse the credibility of the forwarding information is, and the worse the link stability is. Figure 5 shows a comparison of the trust degree of the three algorithms. It can be seen from Fig. 5 that the trust degree of communication tree constructed by TFT algorithm is the best, the trust degree of communication tree constructed by IFT algorithm is about 80%, and the trust degree of nodes is not considered by LFT algorithm, so the trust degree of communication tree is the worst, about 50%.

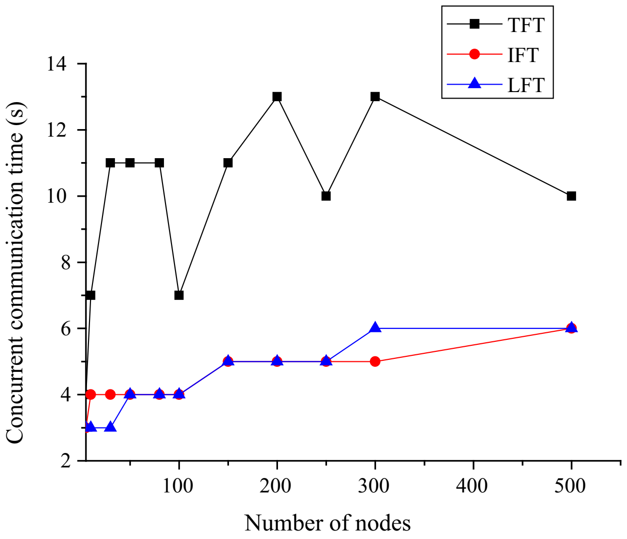


Fig. 4 Comparison of concurrent communication time of three algorithms

5.3 Comparative analysis of communication performance of several algorithms of multifactor blockchain

The communication weight is 10 × 10 weight matrix. For the single-connection communication tree, it is represented by child brother. In the actual communication of blockchain, it is specified that the proportions of communication connection number 3 and 2 are 5:100 and 10:100, respectively. Due to the weak communication forwarding ability of most nodes, the proportions of nodes with 1 and 0 communication connection number are 45:100 and 40:100, respectively. MMWT algorithm is compared with LFT and TFT in terms of communication performance. *Y* axis in Fig. 6 represents concurrent communication time; *Y* axis in

Table 1 Comparison of communication performance of three algorithms

Algorithm	Tree construction rules	Performance evaluation index						Optimization target
		$f(t)$	$f_{AED}(t)$	T_d	U_r	C_d	A_{ls}	
LFT	Number of communication connections first	Excellent	Excellent	Poor	Good	Excellent	Excellent	To shorten the time of concurrent communication
TFT	Trust priority	Poor	Poor	Excellent	Good	Excellent	Poor	Fully consider the stability and reliability of communication
IFT	Considering the number of communication connections and trust	Excellent	Excellent	Good	Excellent	Excellent	Excellent	Considering communication efficiency and reliability

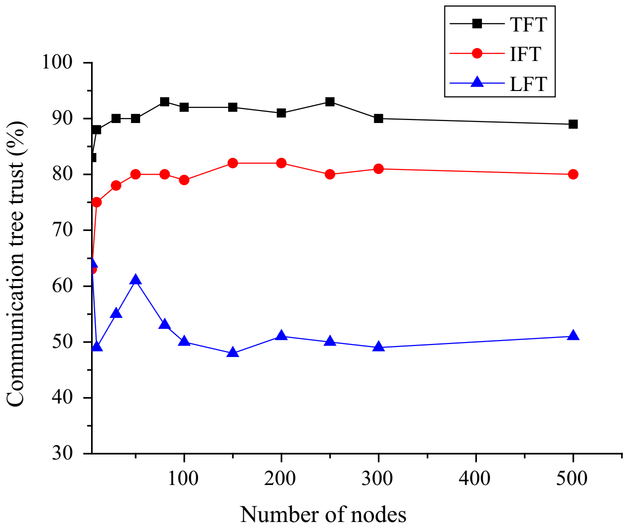


Fig. 5 Comparison of communication tree trust of three algorithms

Figs. 7 and 8 represents average end-to-end delay and the trust degree of the communication tree, respectively. The communication tree constructed by the MMWT algorithm proposed in this chapter is an unequal weight communication tree. As shown in Fig. 4, the concurrent communication time of MMWT algorithm and LFT algorithm increases gently with the increase in the number of nodes, and their values are close to each other, while the concurrent communication time of TFT algorithm is significantly longer than that of MMWT algorithm and LFT algorithm. Therefore, as shown in Fig. 6, the concurrent communication time of MMWT algorithm is sometimes better than that of LFT algorithm. In the case of non-equivalence, the concurrent communication time of the communication tree constructed by LFT algorithm is not necessarily optimal. The reason is that, after the introduction of the communication weight, the concurrent communication time is jointly determined by the number of communication connections and the communication weight, instead of only depending on a single impact factor.

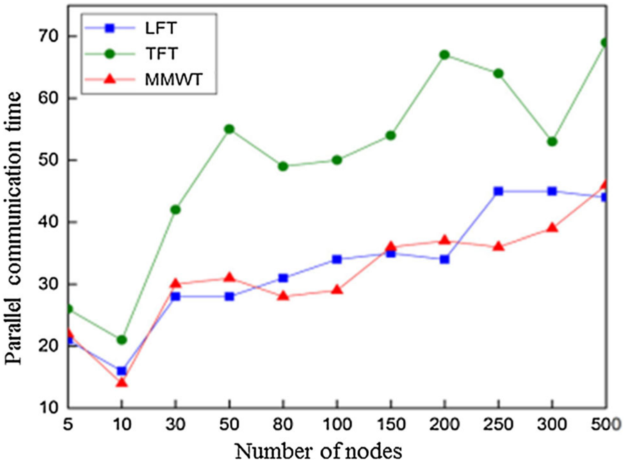


Fig. 6 Parallel communication time

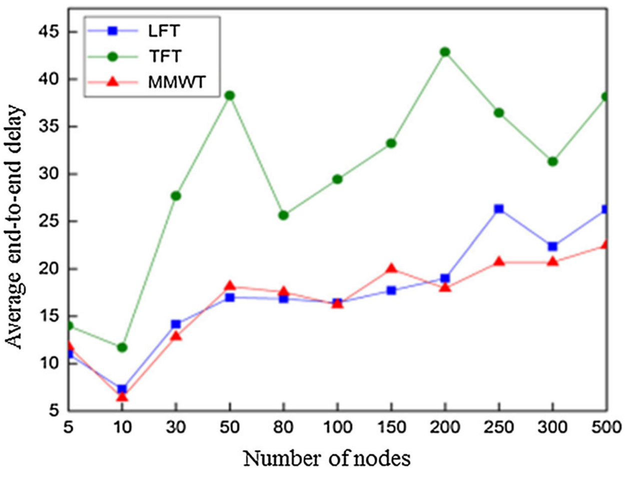


Fig. 7 Average end-to-end delay

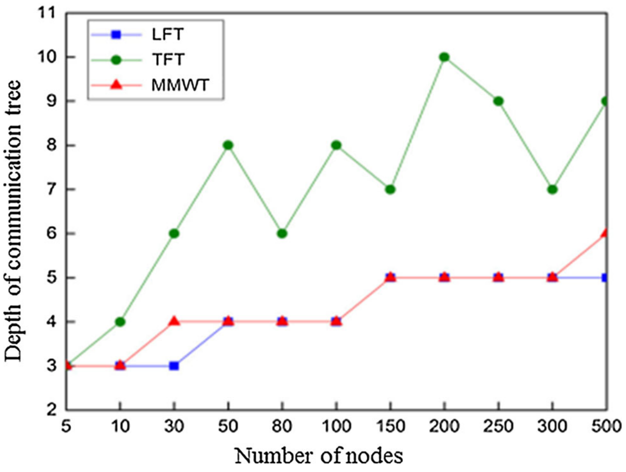


Fig. 8 Depth of communication tree

The average end-to-end delay represents the average time consumed by sending block data from the root node to the target node, and it reflects the average transmission delay of completing the whole blockchain communication. As shown in Fig. 7, the average end-to-end delay of MMWT algorithm and LFT algorithm is obviously better than that of TFT algorithm. The reason is that MMWT algorithm and LFT algorithm give priority to select the nodes with a strong communication ability to improve communication efficiency when constructing communication trees. However, TFT algorithm first selects the nodes with a strong reliability to join the communication tree, so the number of communication connections and the size of communication weights are ignored. TFT algorithm mainly aims at optimizing the reliability and stability of communication link, leading to a larger transmission delay of block data. In most cases, the average end-to-end delay of MMWT algorithm is superior to that of LFT algorithm. With the increase in node *N*, this trend becomes more obvious, because when constructing communication trees, MMWT algorithm additionally considers communication weights, eliminates too large communication weight, and avoids the nodes with a big communication weight from being at the top and upper layer of the communication tree, so as to effectively shorten the average end-to-end communication delay.

The depth of communication tree is also an important indicator for evaluating communication tree algorithm. According to performance evaluation index in 2.3.2, the depth of communication tree constructed by a good algorithm is low, which means that the data have fewer forwarding times on the logical link. For the communication tree with the same number of nodes, if the depth of the communication tree is small, the branches and leaves must be luxuriant. Figure 6 shows that the communication tree constructed by MMWT algorithm and LFT algorithm has a small depth, while that constructed by TFT algorithm has a large depth. Then, the communication trees constructed by MMWT algorithm and LFT algorithm are short and robust, while the communication trees constructed by TFT algorithm are relatively slender. When the communication weight of any two nodes in MMWT algorithm is 1, the research situation of this algorithm will degenerate into the research situation of IFT algorithm in chapter 3. Therefore, IFT algorithm studied in chapter 3 can be regarded as a special case of MMWT algorithm under the condition of constant weight.

6 Conclusion

Based on the multi-connection concurrent communication tree model and the above-mentioned evaluation method of node trust, a blockchain communication algorithm IFT considering node trust is put forward. In this algorithm, the number of communication connections and the trust degree of nodes are considered comprehensively. Moreover, the nodes with strong communication ability and high trust degree are preferred to join the top or upper layer of the communication tree, thereby improving the communication efficiency and reliability of the blockchain. Because of the multi-connection concurrent communication tree model, a multi-connection multifactor communication tree algorithm MMWT based on communication weight is proposed. Considering that the transmission delay of any two nodes in blockchain network is not the same, this paper introduces the concept of communication weights to represent the communication cost between nodes, and comprehensively considers node communication connections, node support degree, and communication weight. First of all, the nodes are ranked according to node communication connections and node support degree from big to small, and then, the malicious nodes and nodes with a large weight are adjusted to the tail of the queue by setting the threshold values of credibility and weight. Based on the above processing, the nodes with a strong communication ability, high credibility, and small weight are located at the top and upper level of the communication tree, thus improving the concurrency of communication and increasing the reliability of blockchain communication. MMWT algorithm, by setting trust threshold and weight threshold, avoids malicious nodes and nodes with large communication weight from being on the top or top of the communication tree, and tries to make high-performance nodes on the top or top of the communication tree, thus improving the efficiency and reliability of blockchain transaction verification. Simulation results and relevant analysis show that, compared with existing algorithms, MMWT algorithm shows a better communication performance in terms of concurrent communication time, average end-to-end delay, concurrency, communication tree credibility, average link pressure, and communication tree depth.

The algorithm studied in this paper belongs to the static routing algorithm, so every blockchain transaction needs to reconstruct the communication tree. Frequent changes to the logical topology of the network will increase the complexity of the operation process and increase the cost. Therefore, in the follow-up study, dynamic routing strategy will be considered to avoid network congestion caused by frequent link changes.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Chang SE, Chen YC, Lu MF (2019) Supply chain re-engineering using blockchain technology: a case of smart contract based tracking process. Technol Forecast Soc Change 144:1–11

Cho SH (2018) A study on analysis of the trend of blockchain by key words network analysis. J Korea Inst Inf Electron Commun Technol 11(5):550–555

Chung YS, Cha JS (2018) The security risk and countermeasures of blockchain based virtual currency trading. J Korea Inst Inf Electron Commun Technol 11(1):100–106

Cocco L, Pinna A, Marchesi M (2017) Banking on blockchain: costs savings thanks to the blockchain technology. Future Internet 9(3):25

Dubey R, Gunasekaran A, Bryde DJ et al (2020) Blockchain technology for enhancing swift-trust, collaboration and resilience within a humanitarian supply chain setting. Int J Prod Res 6:1–18

Goranović A, Meisel M, Fotiadis L et al (2017) Blockchain applications in microgrids an overview of current projects and concepts. In: IECON 2017-43rd annual conference of the IEEE industrial electronics society. IEEE, pp 6153–6158

Grover P, Kar AK, Janssen M et al (2019a) Perceived usefulness, ease of use and user acceptance of blockchain technology for digital transactions–insights from user-generated content on Twitter. Enterp Inf Syst 13:1–30

Grover P, Kar AK, Janssen M et al (2019b) Perceived usefulness, ease of use and user acceptance of blockchain technology for digital transactions–insights from user-generated content on Twitter. Enterp Inf Syst 13(6):771–800

Huckle S, Bhattacharya R, White M et al (2016) Internet of things, blockchain and shared economy applications. Proc Comput Sci 98:461–466

Hussein AF, Arunkumar N, Ramirez-Gonzalez G et al (2018) A medical records managing and securing blockchain based system supported by a genetic algorithm and discrete wavelet transform. Cogn Syst Res 52:1–11

Hussen HM (2018) A blockchain-based service provider validation and verification framework for healthcare virtual organization. UHD J Sci Technol 2(2):24–31

Khezr S, Moniruzzaman M, Yassine A et al (2019) Blockchain technology in healthcare: a comprehensive review and directions for future research. Appl Sci 9(9):1736

Kim SG (2018) A study on the blockchain 2.0 ethereum platform analysis for DApp development. J Korea Inst Inf Electron Commun Technol 11(6):718–723

Meeuw A, Schopfer S, Wortmann F (2019) Experimental bandwidth benchmarking for P2P markets in blockchain managed microgrids. Energy Proc 159:370–375

Mengelkamp E, Notheisen B, Beer C et al (2018) A blockchain-based smart grid: towards sustainable local energy markets. Comput Sci Res Dev 33(1–2):207–214

Muthanna A, Ateya A, Khakimov A et al (2019) Secure and reliable IoT networks using fog computing with software-defined networking and blockchain. J Sens Actuat Netw 8(1):15

Pustišek M, Kos A (2018) Approaches to front-end iot application development for the ethereum blockchain. Proc Comput Sci 129:410–419

Puthal D, Malik N, Mohanty SP et al (2018) Everything you wanted to know about the blockchain: its promise, components, processes, and problems. IEEE Consum Electron Mag 7(4):6–14

Qian Y, Jiang Y, Chen J et al (2018) Towards decentralized IoT security enhancement: a blockchain approach. Comput Electr Eng 72:266–273

Rahman A, Hossain MS, Rahman Z et al (2019) Performance enhancement of the internet of things with the integrated blockchain technology using RSK sidechain. Int J Adv Technol Eng Exp 6(61):257–266

Saberi S, Kouhizadeh M, Sarkis J et al (2019) Blockchain technology and its relationships to sustainable supply chain management. Int J Prod Res 57(7):2117–2135

Sikorski JJ, Haughton J, Kraft M (2017) Blockchain technology in the chemical industry: machine-to-machine electricity market. Appl Energy 195:234–246

Singh S, Ra IH, Meng W et al (2019) SH-BlockCC: a secure and efficient internet of things smart home architecture based on cloud computing and blockchain technology. Int J Distrib Sens Netw 15(4):1550147719844159

Wang J, Gao L, Dong A et al (2017) Blockchain based data security sharing network architecture research. J Comput Res Dev 54:742

Xie J, Tang H, Huang T et al (2019) A survey of blockchain technology applied to smart cities: research issues and challenges. IEEE Commun Surv Tutor 21(3):2794–2830

Yang HK, Cha HJ, Song YJ (2018) Secure identifier management based on blockchain technology in NDN environment. IEEE Access 7:6262–6268

Zeng Z, Li Y, Cao Y et al (2020) Blockchain technology for information security of the energy internet: fundamentals, features, strategy and application. Energies 13(4):881

Zhao G, Liu S, Lopez C et al (2019) Blockchain technology in agri-food value chain management: a synthesis of applications, challenges and future research directions. Comput Ind 109:83–99

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.