Implementation of Improved Raft Consensus Algorithm in IoT Information Security Management

Mingzhen Zhang
School of Artificial Intelligence, Zhengzhou Railway Vocational and Technical College, Zhengzhou, 451460, China

***Abstract*—In the context of the rapid expansion of the Internet of Things, information security management has become particularly crucial. In response to the performance bottleneck of traditional Raft consensus algorithms, this study proposes an improved Raft algorithm that combines density noise spatial clustering algorithm and vote change mechanism, aiming to improve the quantity processing efficiency and consistency of Internet of Things systems in large-scale environments. Firstly, a density noise spatial clustering algorithm is added to the traditional Raft algorithm to partition all consensus nodes into multiple sub clusters. Subsequently, a vote change mechanism is introduced to optimize the leadership election process. Finally, an Internet of Things information security management model is built using the improved Raft algorithm. The results showed that the improved Raft algorithm could complete 500 client requests in just 9.5 minutes of consensus trading time. The log replication accuracy of the management model built using this algorithm under four bandwidth conditions of 0.5Mbps, 5Mbps, 50Mbps, and 500Mbps was as high as 0.98, 0.99, 0.98, and 0.97, respectively. Therefore, the designed consensus algorithm not only has good data processing capabilities, but the model built using this algorithm can also achieve good performance in practical applications.**

Keywords—Blockchain; consensus algorithm; Internet of Things; information; management; raft

I. INTRODUCTION

With the popularization of the Internet of Things (IoT), from smart home to industrial automation, countless devices are connected through the Internet, producing a large amount of data. The value of these data is enormous, but it also raises serious concerns about security and privacy, such as unauthorized data access, data tampering, and device manipulation [1-2]. In this context, a powerful and reliable Information Security Management (ISM) system is needed to protect this data. Consensus algorithms play a crucial role in this process, ensuring that multiple nodes in the network reach consensus on the authenticity and consistency of data without central authority [3]. However, with the surge in the number of IoT devices, traditional consensus algorithms such as Raft face challenges in processing efficiency and scalability [4-5]. Therefore, improving these algorithms and effectively applying them to IoT-ISM has become an urgent issue that needs to be addressed. In the context of the rapid development of the IoT, although the Raft consensus algorithm has been widely used in many fields, its efficiency and scalability in large-scale and highly dynamic environments still have obvious limitations. Most existing research focuses on improving the efficiency and security of consensus algorithms, however, these studies often only focus on the consensus algorithm itself, without involving comprehensive solutions for applying it to IoT-ISM. The aim of this study is to enhance the efficiency and consistency of Raft consensus algorithm for processing large numbers of data nodes in IoT environment by improving it. The objective of this study is to design and implement an improved Raft algorithm combining Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Vote Change Mechanisms (VCM) to improve the processing efficiency and data consistency of IoT-ISM. The importance of this paper is that it can significantly improve the data processing capabilities and security of IoT devices, providing practical solutions and theoretical basis for the sustainable development of IoT technology.

The main contribution of this study is to propose an improved Raft consensus algorithm combining DBSCAN and VCM. This improvement has the dual benefit of optimizing data processing efficiency and system consistency in the IoT-ISM, as well as corroborating the significant effect of the algorithm on improving log replication accuracy and reducing Consensus Transaction Time (CTT). In addition, this study also constructs an improved Raft algorithm based on the IoT-ISM model, which demonstrates superior performance in different network bandwidths and complex environments. Consequently, this study not only offers an efficacious technical solution for the ISM of the IoT, but also provides a novel perspective and empirical data to support the research of consensus algorithms.

The structure of this paper is as follows: The first part reviews the relevant work, discusses the existing consensus algorithm and its application in the IoT-ISM, and points out the shortcomings of the existing research. The second part details the design of the improved Raft consensus algorithm, including the integration of DBSCAN and the introduction of VCM. The third part describes the construction process of the IoT-ISM model based on the improved Raft algorithm, and shows the specific framework of the model. The fourth part verifies the performance of the improved algorithm and model through experiments, including the evaluation of key indicators such as CTT, log replication accuracy and system adaptability. Finally, the paper summarizes the full text and discusses the theoretical and practical significance of the research results. At the same time, the future research direction is prospected.

II. RELATED WORKS

To improve the correctness and immutability of all transactions in blockchain, many experts have optimized consensus algorithms. Rong B et al. explored the optimization strategy of Raft consensus algorithm in the rapid growth of distributed clusters and the rapid decline of throughput, and proposed a federal restructuring committee Raft consensus algorithm. This algorithm was based on federated reconstruction technology, which trained, updated, and evaluated the feature dataset model of Raft nodes, selected nodes with better performance, constructed a committee mechanism, and improved the quality and speed of elections. At the same time, to address the inconsistency and security issues in federated aggregation, a semi-asynchronous buffering mechanism and defense strategies against malicious node attacks have been designed. The effectiveness of this algorithm has been validated in consensus clusters [6]. Raft consensus algorithm is a key technology for state replication in distributed systems. The state updates in Raft consensus algorithm are influenced by the leader node, and the system response time is also affected by the delay between nodes. Choumas K et al. proposed a mathematical model to estimate the waiting time range that affects the probability of leadership elections in Raft consensus algorithm, aiming to reduce the expected response time of the system. The performance of the model was validated through the open-source Raft testing platform in the article, which showed that optimizing the interval time can improve the probability of selecting leader nodes in the Raft consensus algorithm, thereby optimizing the node selection results [7]. To overcome the scalability limitations and high cost issues of blockchain applications in IoT systems, Guo H et al. proposed a consensus protocol algorithm with a hierarchical structure and location awareness, referred to as LH-Raft. It confirmed the scalability of LH-Raft in large-scale IoT applications. This algorithm could effectively reduce communication costs, consensus latency, and protocol time [8]. Aiming at the problems of vote falsification and malicious election of candidate nodes existing in the traditional Raft consensus algorithm, Tian S et al. proposed a new consensus algorithm combining zero-trust mechanism and secret sharing technology, which was recorded as VSSB-Raft. This algorithm achieved zero-trust through monitoring nodes and secret sharing algorithms, without relying on hidden trust between nodes. It also used the SM2 signature algorithm to strengthen authentication before data usage, ensuring data security. In addition, by introducing named data network, the communication mode between nodes was redesigned to ensure the quality of node communication. The results demonstrated that VSSB-Raft consensus algorithm achieved high throughput and low consensus delay while maintaining algorithm complexity, effectively improving system security and efficiency [9].

IoT environments typically involve a large number of devices and sensors that generate, collect, and exchange large amounts of data. In this environment, ensuring the security, privacy, and integrity of data is a major challenge. Khan A et al. proposed a novel blockchain architecture BHI-IoT for electronic health data security. This architecture aimed to enhance network resources and trust in industrial IoT by optimizing data management and distributed layered architecture of medical wireless sensor networks. BHI-IoT adopted NuCypher threshold re-encryption mechanism to protect data, utilizing customized lightweight blockchain and digital signatures with multiple proof of work and multiple proof of rights to reduce resource consumption and storage burden. This framework could effectively improve the security and efficiency of IoT systems in the electronic health industry [10]. Hasan N et al. proposed a blockchain driven network physical system. This system aimed to achieve ISM and lightweight data management in IoT systems by utilizing smart contracts and peer-to-peer databases. The system solved data storage and transmission problems through the integration of private blockchain and intelligent device micro-controllers, and demonstrated the application effect of the system in food supply chain traceability [11]. In the medical field, IoT faced challenges in patient information privacy and data integrity. ElRahman S A et al. proposed an IoT-Edge framework that integrates blockchain technology to securely exchange data, ensure data integrity and privacy. This framework allowed IoT devices to remotely monitor patient status while ensuring secure transmission and storage of patient data. User friendly system design provided necessary tools for information integrity and confidentiality. After simulation testing, the framework has proven its feasibility in medical applications [12]. Aiming at the trust management challenges caused by limited bandwidth and long latency in underwater IoT, Jiang J et al. proposed a dispute adjudication method to deal with trust recommendation conflicts, and developed a new trust management mechanism based on this. The mechanism included three stages: trust calculation, trust recommendation and trust evaluation. By collecting trust evidence such as packet transmission rate and combining with incentive mechanism based on prisoner's dilemma, the neighbor was encouraged to participate in trust recommendation. Simulation results showed that this mechanism was superior to existing studies in terms of accuracy and robustness [13].

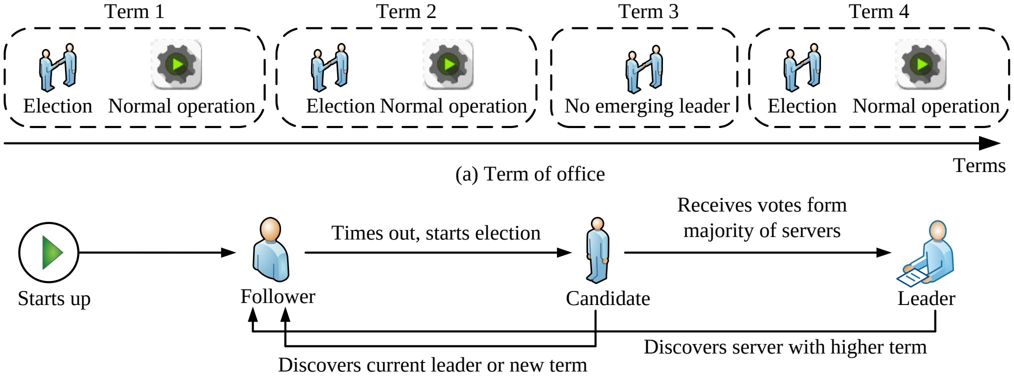
In summary, consensus algorithms play a crucial role in distributed computing and blockchain technology, and many experts have conducted a series of optimization studies on consensus algorithms. Currently, most research only focuses on the consistency and integrity issues of consensus algorithms, and few experts have conducted joint research on consensus algorithms and IoT's ISM. Based on this background, this study aims to optimize Raft consensus algorithm by combining clustering algorithms and VCM, and use optimization algorithms to build a complete IoT-ISM model, aiming to further improve the storage and privacy issues of IoT big data.

III. IoT-ISM Based on Improved Consensus Algorithm

To ensure the privacy and security of IoT information, this study first optimizes the traditional Raft consensus algorithm by using the DBSCAN algorithm and VCM to change its election mechanism. By using a multi-cluster structure to increase the number of leaders and distribute the load of the entire consensus system, the efficiency of the algorithm is improved. On this basis, a complete IoT-ISM model is constructed using an improved Raft consensus algorithm.

A. Improved Raft Consensus Algorithm Design Integrating DBSCAN and VCM

In Raft consensus algorithm, all nodes in the cluster are divided into three roles: leader, follower, and candidate. At the beginning, all nodes are followers. If the follower does not receive information from the leader within a certain period of time, it will become a candidate and start a new round of elections. Once a leader is selected, the leader node will be responsible for managing client requests and copying them as log entries to other nodes. It is only when these log entries are stored by the majority of nodes that the operations in question can be committed and applied to the state machine of each node. In Raft consensus algorithm, term is a very important concept. Term is a logical clock used in Raft consensus algorithm to distinguish different time periods, mainly used to solve the problems of leader election and log replication in distributed systems. Fig. 1 shows the term structure and node transition process.


Term 1
Term 2
Term 3
Term 4
Election
Normal operation
Election Normal operation
No emerging leader
Election
Normal operation
Terms
Starts up
Follower
Times out, starts election
Candidate
Receives votes form majority of servers
Leader
Discovers server with higher term
Discovers current leader or new term

(a) Term of office

(b) Three state transitions of nodes

Fig. 1. Term structure and node transition diagram.

In Fig. 1 (a), each term represents a period of time, such as Term1, Term2, Term3, etc. During this period, a node in the cluster will act as the leader to coordinate the replication of logs. The term of office is identified by consecutive integers, and the term number increases every time a new leader election occurs. Each leader node has a certain period of tenure, during which the leader node will perform leadership actions. During a term, only one leader will be elected. If a leader loses contact with most nodes for some reason, a new term will begin and a new round of leader elections will be held. In Fig. 1 (b), all nodes of Raft will default to the follower state at startup. If the follower does not receive the leader's information within the scheduled time, they will become candidates and initiate a new round of elections by increasing their term number and requesting other nodes to vote. If the candidate receives a majority of votes, they become the leader. If a candidate receives information from the current leader or does not receive a majority of votes, the candidate will either retreat to their followers or start a new round of elections. Considering that traditional Raft consensus algorithm heavily relies on the performance of the leader node, the performance of traditional Raft consensus algorithm will be affected as the number of follower nodes continues to increase. Therefore, this study proposes an improved Raft consensus algorithm that integrates DBSCAN and VCM, and the improved algorithm is referred to as DBSCAN-Raft. The flowchart is Fig. 2.

In Fig. 2, the DBSCAN-Raft algorithm first uses clustering methods to divide all consensus nodes into multiple sub clusters. Each sub-cluster elects a sub-leader through Raft, with the remaining nodes serving as followers. These sub-leaders then form the main cluster and execute the Raft algorithm as a whole. To avoid deadlock caused by competition among multiple candidates, VCM is introduced. When no candidate receives more than half of the votes, the candidate with the highest number of votes will become the leader. By using a multi-cluster structure to increase the number of leaders and distribute the load of the entire consensus system, algorithm efficiency can be improved.

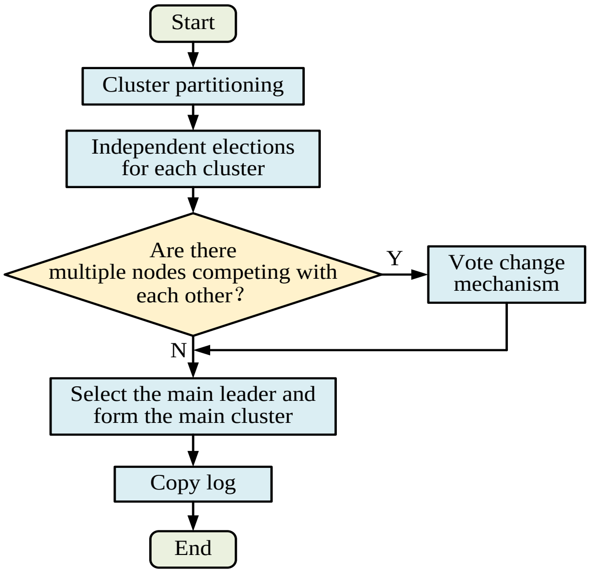

Start
Cluster partitioning
Independent elections for each cluster
Are there multiple nodes competing with each other?
Y
Vote change mechanism
N
Select the main leader and form the main cluster
Copy log
End

Fig. 2. Flowchart of the operation of DBSCAN-Raft.

In cluster analysis, it is assumed that there is a dataset X with dimension D and n data points. The expression of X is Eq. (1).

$$X=\left\{x_i \,\middle|\, x_{i,1},x_{i,2},x_{i,3},\cdots x_{i,j}\right\} \tag{1}$$

In Eq. (1), x_i represents the data in the dataset, $i\in[1,n]$. j represents the dimension in which each data is located, $j\in[1,D]$. According to the similarity between data points, the data is divided into groups as shown in Eq. (2).

$$C=\left\{C_1,C_2,\cdots,C_k\right\} \tag{2}$$

In Eq. (2), C represents the data group, and k represents the category of the data group. When dividing data groups, the constraints in Eq. (3) need to be met.

$$\begin{cases} C_p \neq \varnothing \\ C_1 \cup C_2 \cup, \cdots, \cup C_k = X \\ C_P \cap C_q \neq \varnothing \quad p \neq q \end{cases} \tag{3}$$

In Eq. (3), p and q represent two different groups, $p, q \in [1, k]$, respectively. There are a total of three constraints that need to be met in Eq. (3). Firstly, each group is not an empty set and must contain at least one data. Secondly, the union of all grouped sets is the entire dataset. Thirdly, there is no inclusion relationship between each group, meaning that one data belongs only to one group. In DBSCAN, assuming the domain of data point a is $N_\varepsilon(a)$, its expression can be expressed as Eq. (4).

$$N_\varepsilon(a) = \{ b \in D_s \,|\, d(a,b) \leq \varepsilon \} \tag{4}$$

In Eq. (4), D_s represents the dataset. b represents another data point. $d(a,b)$ represents the distance between a and b. ε represents the field. The calculation of density is Eq. (5).

$$\rho(a) = |N_\varepsilon(a)| \tag{5}$$

In Eq. (5), $\rho(a)$ represents the density value of a. The density value reflects the number of data points included in the ε domain. The expression for the core point is Eq. (6).

$$\rho(a) \geq \text{MinPts} \tag{6}$$

In Eq. (6), MinPts represents the density threshold. When equation (6) holds, then a will become the core point. The boundary points is expressed as Eq. (7).

$$\begin{cases} \rho(a) < \text{MinPts} \\ a \in N_\varepsilon(b) \end{cases} \tag{7}$$

In Eq. (7), when a satisfies the conditions in Eq. (7), a becomes a non-core point, i.e. a boundary point. Due to $a \in N_\varepsilon(b)$, b will become the core point at this time. The expression of noise points is Eq. (8).

$$\rho(a) \neq \text{MinPts} \tag{8}$$

In Eq. (8), when a is neither the core point nor the boundary point, it is recorded as a noise point. Assuming that DBSCAN has a set of C_l that satisfies the clustering conditions as shown in Eq. (9).

$$\begin{cases} C_l \in D_s \\ C_l \neq \varnothing \\ a, b \in C_l \end{cases} \tag{9}$$

In Eq. (9), when C_l is non-empty and belongs to dataset D_s, two conditions will be met. Firstly, if $a \in C_l$ and the density from a to b can reach, then there exists $b \in C_l$. Secondly, if $a, b \in C_l$, it indicates that a and b are connected in density. Using kernel density estimation method to optimize the value of ε, the calculation is Eq. (10).

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i) \tag{10}$$

In Eq. (10), $\hat{f}_h(x)$ represents the probability density function estimated using sample data at data x. $K(\cdot)$ represents the kernel function, which is generally represented using a standard Gaussian function [14]. h represents the bandwidth parameter used to control the width of the kernel function, and its determination formula is Eq. (11).

$$MISE(h) = \frac{\int K^2(x)dx}{nh} + \frac{h^4 \sigma^4 \int [f''(x)]^2 dx}{4} + \left(\frac{1}{nh} + h^4 \right) \tag{11}$$

In Eq. (11), $MISE(h)$ represents the mean square error value of kernel density estimation. $K^2(x)dx$ represents the square integral of $K(\cdot)$, used to measure the smoothing characteristics of the kernel function itself. σ represents the standard deviation of the data. $\int [f''(x)]^2 dx$ represents the integral of the square of the second derivative $f''(x)$ of the probability density function. Using the rule of thumb to optimize Eq. (11), the optimal solution formula for ε is ultimately obtained as shown in Eq. (12).

$$h' = \left(\frac{4}{3n} \right)^{\frac{1}{5}} \sigma \tag{12}$$

In Eq. (12), h' represents the optimal solution ε obtained. By using the above formula, node clustering can be completed, and the topology diagram of node clustering is Fig. 3.

In Fig. 3, after completing cluster partitioning, the core points and their associated boundary points form a separate cluster, and Raft consensus algorithm is used for consensus operation. For noise points that have not been classified into any specific cluster, this study incorporates them into the main cluster to participate in the consensus process. In addition, during the voting stage of leader elections, to solve the situation of deadlock among multiple candidate nodes, this study designs a VCM strategy, through which the candidate node with the most votes can obtain the leader position.

B. Construction of IoT-ISM Model Based on DBSCAN-Raft

After optimizing the Raft algorithm using DBSCAN and VCM, this study further builds an IoT-ISM model based on the improved DBSCAN-Raft. It aims to protect user data privacy and store user information through this model. Fig. 4 shows the constructed IoT-ISM model framework.

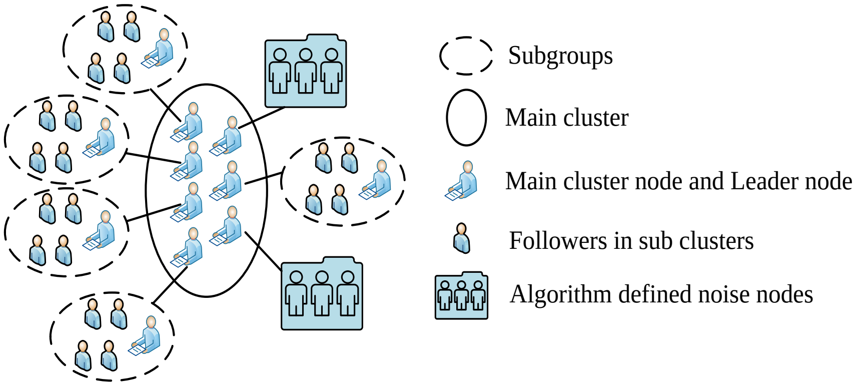

Subgroups
Main cluster
Main cluster node and Leader node
Followers in sub clusters
Algorithm defined noise nodes

Fig. 3. Node clustering topology.

In Fig. 4, a complete IoT-ISM model mainly consists of a Data Processing Layer (DPL), Clustering and Annotation Layer (C/AL), Consensus Mechanism Layer (CML), Secure Communication Layer (SCL), Main Cluster Management Layer (MCML), Decision and Response Layer (D/RL), and Monitoring and Maintenance Layer (M/ML) [15-16]. DPL is segmented into data collection and pre-processing. Firstly, it deploys device nodes on IoT devices, collects data through the devices, processes input data from sensors, removes noise and outliers, and standardizes data formats. Finally, meaningful information is extracted from the raw data as features. C/AL mainly clusters preprocessed data to determine core points, boundary points, and noise points. In addition, this layer also needs to annotate the data points in the clustering results to identify key and non key data. CML executes DBSCAN-Raft and selects the leading node to complete log replication. SCL aims to ensure the encryption and security of data during transmission, ensuring that only authorized nodes can participate in the consensus process. MCML aims to incorporate all sub-leader nodes into the main cluster and elect the main leader through DBSCAN-Raft. The main leader then coordinates with each sub-leader node to synchronize the status information of each sub-cluster. D/RL will automatically execute business logic based on consensus results, and execute corresponding security measures based on decisions, such as data backup, recovery, and intrusion response. M/ML will monitor the real-time operation status of the IoT environment, regularly maintain and update the system to adapt to environmental changes and new security threats [17].

Within the consensus layer composed of DBSCAN-Raft, data is stored between nodes through logs. The leader node is responsible for various requests initiated by the client and carries out a series of log copying and confirmation processes. The core purpose of this process is to ensure that the data of all nodes in the cluster remains synchronized. When a client initiates a transaction request, it is first processed by the leader node, which is then responsible for propagating these log entries to the follower node. To maintain data consistency in the system, the leader node will follow two basic principles: First, it will not delete any client request records, and second, the follower node will only synchronize log data from the leader node. The process of log replication is Fig. 5.

The log replication in Fig. 5 needs to follow the following steps. Firstly, the request initiated by the client contains pending commands to be executed. Next, the leader adds the command as a new log entry to their log file and broadcasts it to other follower nodes through remote procedure call communication, requesting them to copy the entry. After the follower node completes replication, it will provide feedback to the leader on its replication status. Once the leader receives confirmation of successful replication from most nodes, it can be considered that the log entry replication is complete [18-19]. The leader then updates their state machine and reports success to the client. On the contrary, if no majority response is received, a failure is reported to the client [20-21]. The node state changes under normal log replication and the log replication state under conflict state are shown in Fig. 6.

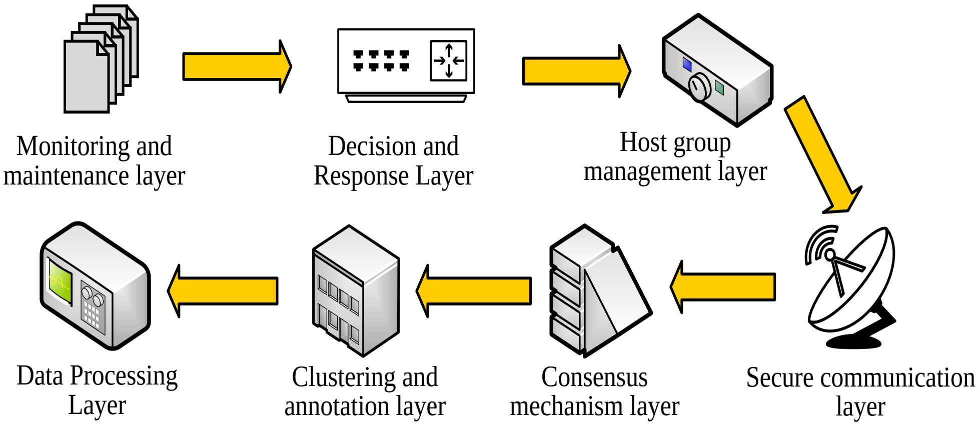

Monitoring and maintenance layer
Decision and Response Layer
Host group management layer
Data Processing Layer
Clustering and annotation layer
Consensus mechanism layer
Secure communication layer

Fig. 4. IoT-ISM model.

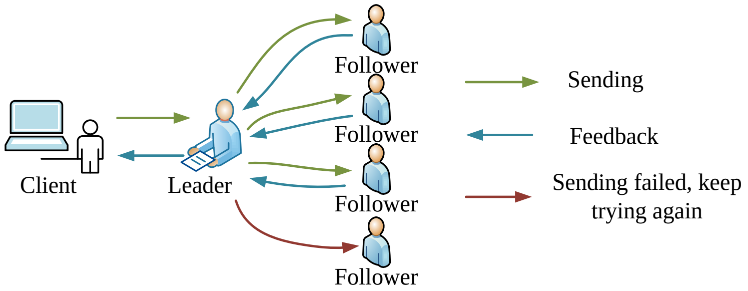


Fig. 5. Log replication flowchart.

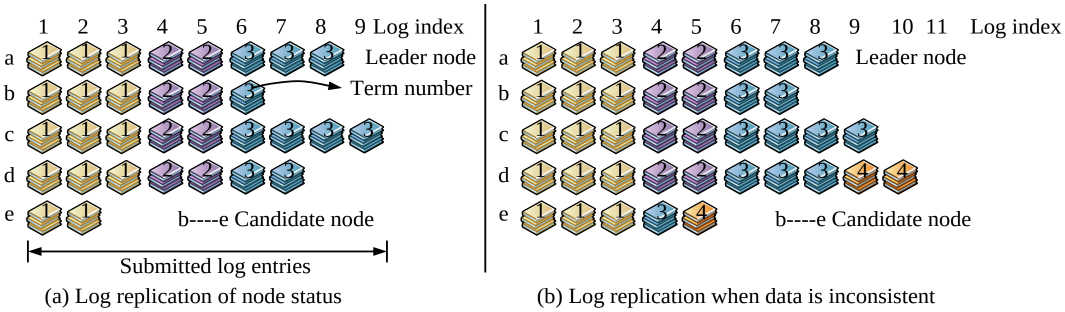

1 2 3 4 5 6 7 8 9 Log index
a
b
c
d
e
Leader node
Term number
b----e Candidate node
Submitted log entries
(a) Log replication of node status
1 2 3 4 5 6 7 8 9 10 11 Log index
Leader node
b----e Candidate node
(b) Log replication when data is inconsistent

Fig. 6. Log replication in node state and conflict state.

Fig. 6 (a) and (b) show the node state changes under normal and abnormal log replication states. In Fig. 6 (a), nodes a and c have the longest logs and are eligible to become leaders, while nodes b and d are synchronizing logs, while node e stops operating due to anomalies. In this consensus model, the log with index 7 has been consensus among three nodes, and the number of confirmed nodes exceeds more than half of the nodes in the cluster [22-23]. Therefore, log entries below number 7 are considered as consensus and cannot be modified. In 6 (b), the logs of each node may not be consistent with the logs of the new leader. The task of a leader is to restore consistency by having followers abandon conflicting logs and synchronize their logs. The newly selected leader first aligns with the logs of other nodes. If a mismatch is found, consistency checks are performed in the additional log request. If it fails, the log number is not updated and is gradually rolled back to find a matching point. Once it is found, followers will remove the conflict log, adopt the leader's log, and reach a consensus. In the example shown in Fig. 6 (b), node a as the leader does not need to change, node b remains in its current state, node c deletes the last log entry, node d deletes two logs with a term of 4, and node e deletes one log with a term of 4. After completing their respective deletion tasks, each node will synchronize the remaining logs of the leader to ensure data consistency and system security.

IV. Performance Testing of DBSCAN-Raft and Analysis of the Application Effect of ISM Model

To demonstrate the effectiveness of the designed Raft consensus algorithm and ISM models, this study first tests the performance of DBSCAN-Raft. On this basis, the application effect of the ISM model built using DBSCAN-Raft in practical problems is further verified.

A. DBSCAN-Raft Performance Testing

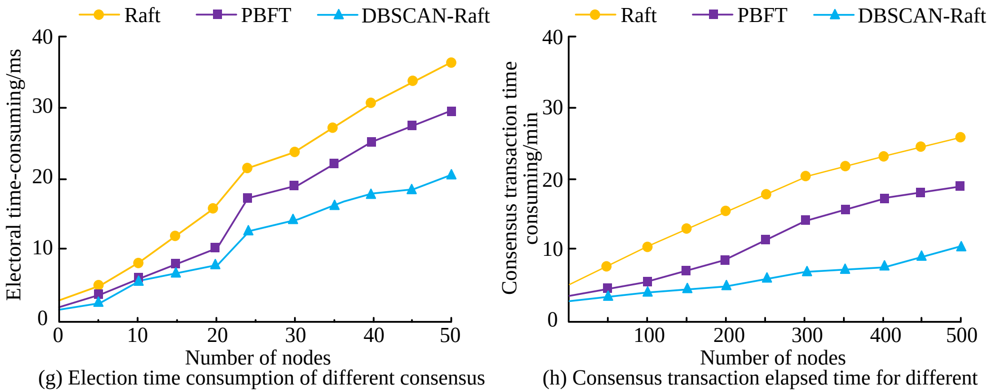
To ensure that all consensus algorithms can be tested in the same experimental environment and effectively avoid experimental errors, the experimental testing environment shown in Table I is constructed in this study.

TABLE I. Experimental Test Environment

Experimental equipment	Value
CPU	Intel Core i9-10900K
GPU	NVIDIA RTX 3080
Memory	4GB RAM 100GB SSD
Graphics Memory	Ubuntu14.04 64-bit
Systems	Windows 10 Python 3.8
Software	Hyperledger Fabric and Caliper

Table I provides the specific settings of the experimental testing environment. The experiment involves setting up 500 client requests and requesting consensus from the cluster at a speed of 200tps. Traditional Raft and Practical Byzantine Fault Tolerance (PBFT) are used as comparative algorithms to obtain the election and CTT of Raft, PBFT, and DBSCAN-Raft under the same experimental conditions, as shown in Fig. 7.

Fig. 7 (a) and 7 (b) show the Election Elapsed Time (EET) and CTT of the three algorithms, respectively. In Fig. 7 (a), as the Number of Nodes (NN) gradually increases from 0 to 50, the EET of Raft, PBFT, and DBSCAN-Raft all show a continuous upward trend. Compared to PBFT and DBSCAN-Raft, Raft has the largest increase in amplitude. When the NN is 50, the EET of Raft, PBFT, and DBSCAN-Raft are 36.8ms, 28.9ms, and 19.7ms, respectively. In Fig. 7 (b), when Raft, PBFT, and DBSCAN-Raft finally completes 500 client requests, the CTTs used are 26.6 minutes, 18.4 minutes, and 9.5 minutes, respectively.



(g) Election time consumption of different consensus algorithms

(h) Consensus transaction elapsed time for different consensus algorithms

Fig. 7. EET and CTT for different consensus algorithms.

Fig. 8 shows the throughput of Raft, PBFT, and DBSCAN-Raft when processing different NNs. When NN increases from 10 to 100, the throughput values of Raft, PBFT, and DBSCAN-Raft all decrease, but the decrease in DBSCAN-Raft is the smallest. When the NN is 10/100, the throughput sizes of Raft, PBFT, and DBSCAN-Raft are 23.6tps/6.9tps, 27.1tps/11.4tps, and 28.8tps/17.5tps, respectively.

Fig. 9 (a) and 9 (b) show the energy consumption and stability curves of the three consensus algorithms, respectively. In Fig. 9 (a), when NN increases from 0 to 100, the energy consumption values of Raft, PBFT, and DBSCAN-Raft will fluctuate continuously in different ranges. Compared to Raft and PBFT, DBSCAN-Raf has the smallest range of energy consumption fluctuations, within 20J. The energy consumption fluctuations of Raft and PBFT are within 50J and 40J, respectively. In Fig. 9 (b), the three algorithms gradually reach a stable state after multiple training. The training times for Raft, PBFT, and DBSCAN-Raft to reach a stable state are 74, 46, and 31, respectively.

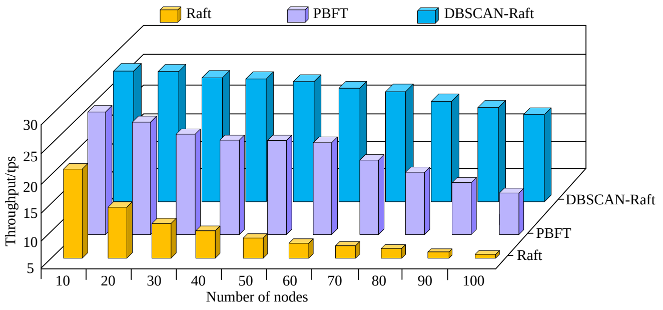
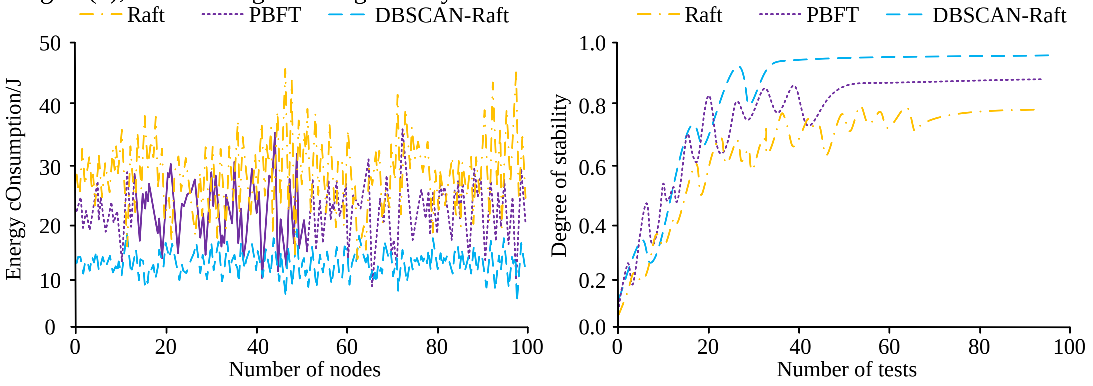


Fig. 8. Throughput of different consensus algorithms.



(a) Energy consumption of different consensus algorithms

(b) Stability of different consensus algorithms

Fig. 9. Energy consumption and stability of different consensus algorithms.

Table II presents the fitness values of three consensus algorithms under extremely low bandwidth (<1Mbps), low bandwidth (1-10Mbps), medium bandwidth (10-100Mbps), and high bandwidth (>100Mbps). The fitness values of Raft under four bandwidth conditions are 0.93, 0.87, 0.82, 0.78, PBFT is 0.96, 0.91, 0.88, 0.82, and DBSCAN-Raft is 0.92, 0.95, 0.96, and 0.94, respectively.

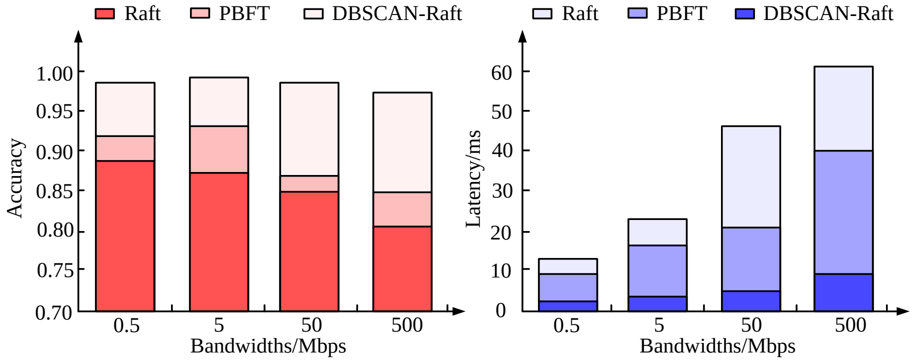
B. Analysis of the Application Effect of Iot-ISM Model

Three different IoT-ISM models are constructed using Raft, PBFT, and DBSCAN-Raft algorithms. The application performance of the log replication module in the model is tested, and the accuracy and latency of log replication for three ISM models are shown in Fig. 10.

Fig. 10 (a) and 10 (b) show the log replication accuracy and latency of three ISM models under different bandwidths, respectively. In Fig. 10 (a), when the actual network bandwidth is 0.5Mbps, 5Mbps, 50Mbps, and 500Mbps, the ISM model designed by DBSCAN-Raft performs the best in log replication accuracy and has the lowest latency. The replication accuracy is as high as 0.98, 0.99, 0.98, and 0.97, with delays of 1.89ms, 3.04ms, 4.72ms, and 6.95ms, respectively. The comparison results in Fig. 10 show that DBSCAN-Raft model has better log replication accuracy and response time than Raft and PBFT model at low bandwidth. This is because the DBSCAN-Raft algorithm performs better in bandwidth-constrained environments by optimizing node election and clustering management, reducing unnecessary data transfers and re-elections. When the bandwidth is increased to 50Mbps and 500Mbps, the performance of all models improves, but DBSCAN-Raft still maintains the highest accuracy and low latency. This indicates that DBSCAN-Raft algorithm can effectively utilize high bandwidth and reduce data loss and errors when processing large amounts of data transmission, thus maintaining high operational efficiency and system stability. It indicates that DBSCAN-Raft algorithm is particularly suitable for environments with large bandwidth variations and can maintain high performance under different network conditions. For the unstable network environment common in IoT applications, the ISM model using DBSCAN-Raft algorithm can provide more reliable and efficient data processing capabilities.

TABLE II. ADAPTATION OF DIFFERENT CONSENSUS ALGORITHMS UNDER DIFFERENT BANDWIDTHS

Type	Bandwidth range	Adaptation value
Raft	<1Mbps	0.93
	1~10Mbps	0.87
	10~100Mbps	0.82
	>100Mbps	0.78
PBFT	<1Mbps	0.96
	1~10Mbps	0.91
	10~100Mbps	0.88
	>100Mbps	0.82
DBSCAN-Raft	<1Mbps	0.92
	1~10Mbps	0.95
	10~100Mbps	0.96
	>100Mbps	0.94



(a) Log replication accuracy for different models at different bandwidths

(b) Log replication delay for different models at different bandwidths

Fig. 10. Accuracy and latency of log replication for three ISM models.

Table III shows the performance of three management models obtained by selecting two network types, Personal Area Network (PAN) and Wireless Local Area Network (WLAN), under different network types.

TABLE III. PERFORMANCE OF DIFFERENT MANAGEMENT MODELS IN TWO NETWORK TYPES

Type	Performance evaluation index	Raft	PBFT	DBSCAN-Raft
PAN	Packet loss rate/%	0.106	0.053	0.012
	Throughput/bps	108	132	196
	Transmission delay/ms	1.25	0.68	0.15
WLAN	Packet loss rate/%	0.068	0.039	0.008
	Throughput/bps	121	164	238
	Transmission delay/ms	0.87	0.34	0.02

Table III presents the packet loss rate, throughput, and transmission delay of three models under two network types. Both in PAN and WLAN, DBSCAN-Raft has lower packet loss rate and transmission delay values than Raft and PBFT, and its throughput is greater than Raft and PBFT. Among them, DBSCAN-Raft performs better in WLAN, with a packet loss rate as low as 0.008%, a transmission delay as low as 0.02ms, and a throughput of up to 238bps.

V. DISCUSSION

Aiming at the limitations of Raft consensus algorithm in processing efficiency and scalability in IoT environment, an improved Raft algorithm combining DBSCAN and VCM is designed to improve the quantity processing efficiency and data consistency of IoT system. Compared with literature [24], although the system proposed in literature [24] performs well in enhancing data security, scalability and efficiency still need to be improved when dealing with large-scale nodes and highly dynamic network environments. In this study, the improved DBSCAN-Raft algorithm not only optimizes the coordination mechanism between nodes, but also effectively reduces election conflicts through the VCM, making it more stable and efficient under dynamic network conditions. In addition, Showkat and Qureshi have extensively discussed the security architecture of IoT through blockchain technology in the literature [25], which provides a variety of security strategies but lacks specific algorithm implementation and performance testing. This study not only proposed the specific algorithm design, but also verified the effectiveness of the algorithm through the actual simulation test. The results showed that the improved Raft algorithm had excellent performance in log replication accuracy. For example, under the bandwidth conditions of 0.5Mbps, 5Mbps, 50Mbps and 500Mbps, the log replication accuracy reached 0.98, 0.99, 0.98 and 0.97 respectively, which was significantly better than the traditional Raft algorithm. At the same time, the CTT of the algorithm was only 9.5 minutes when processing 500 client requests, which was much faster than the traditional method. The improvement of these performance indicators not only showed the advanced nature of DBSCAN-Raft algorithm in theory, but also showed its high efficiency and high reliability in practical applications. In addition, this study also found that through the VCM, the DBSCAN-Raft algorithm can effectively maintain the stability and response speed of the system even in the network environment with intense node competition.

In summary, the proposed DBSCAN-Raft algorithm provides a new idea and solution for solving the performance bottleneck of consensus algorithm in the IoT environment. Future research could further investigate the impact of different types of IoT devices and network conditions on the consensus algorithm, as well as the potential for enhancing the algorithm's efficiency and scalability while maintaining security.

VI. Conclusion

To address the ISM challenge of massive IoT data, this study optimized the traditional Raft consensus algorithm and proposed a new DBSCAN-Raft algorithm, which was then used to build an IoT-ISM model. Experiments have shown that DBSCAN-Raft performed better than Raft and PBFT in indicators such as EET, CTT, and throughput. When NN was 50, the EET of DBSCAN-Raft was 19.7ms, and when NN was 500, the CTT of DBSCAN-Raft was 9.5min. When NN increased from 10 to 100, the throughput of DBSCAN-Raft decreased from 28.8tps to 17.5tps, and the energy consumption fluctuated consistently within 20J, with a much smaller decrease than Raft and PBFT. In addition, DBSCAN-Raft had the highest fitness values at low bandwidth, medium bandwidth, and high bandwidth. The model built using DBSCAN-Raft could achieve log replication accuracy of 0.98, 0.99, 0.98, and 0.97 at four bandwidth values of 0.5Mbps, 5Mbps, 50Mbps, and 500Mbps, respectively. Moreover, the model could achieve a packet loss rate of 0.008%, a transmission delay of 0.02ms, and a throughput of 238bps in WLAN. Considering the energy limitations and computing power of IoT devices, future research should further explore the impact of different types of IoT devices and network conditions on consensus algorithms.

References

[1] Chanal P M, Kakkasageri M S. Security and privacy in IoT: a survey. Wireless Personal Communications, 2020, 115(2): 1667-1693.

[2] Mokayed H, Quan T Z, Alkhaled L, and Sivakumar V. Real-time human detection and counting system using deep learning computer vision techniques. Artificial Intelligence and Applications, 2023, 1(4): 221-229.

[3] Kwak J Y, Yim J, Ko N S, Kim S M. The design of hierarchical consensus mechanism based on service-zone sharding. IEEE Transactions on Engineering Management, 2020, 67(4): 1387-1403.

[4] Liu Y, Wang J, Yan Z, Wan Z, Jantti R. A survey on blockchain-based trust management for Internet of Things. IEEE Internet of Things Journal, 2023, 10(7): 5898-5922.

[5] Jin H S, Kim D O, Kim Y C, Oh J T, Kim K Y. Trend Analysis of High-Performance Distributed Consensus Algorithms. Electronics and Telecommunications Trends, 2022, 37(1): 63-72.

[6] Rong B, Zheng Z. FRCR: Raft Consensus Scheme Based on Semi Asynchronous Federal Reconstruction. IEEE Transactions on Network and Service Management, 2022, 19(4): 3822-3834.

[7] Choumas K, Korakis T. On Using Raft Over Networks: Improving Leader Election. IEEE Transactions on Network and Service Management, 2022, 19(2): 1129-1141.

[8] Guo H, Li W, Nejad M. A hierarchical and location-aware consensus protocol for iot-blockchain applications. IEEE Transactions on Network and Service Management, 2022, 19(3): 2972-2986.

[9] Tian S, Bai F, Shen T, Zhang C, Gong B. VSSB-Raft: A Secure and Efficient Zero Trust Consensus Algorithm for Blockchain. ACM Transactions on Sensor Networks, 2024, 20(2): 1-22.

[10] Khan A A, Bourouis S, Kamruzzaman M M, Hadjouni M, Shaikh Z A, Laghari A A, Elmannai H, Dhahbi S. Data Security in Healthcare Industrial Internet of Things with Blockchain. IEEE Sensors Journal, 2023, 23(20): 25144-25151.

[11] Hasan N, Chaudhary K, Alam M. A novel blockchain federated safety-as-a-service scheme for industrial IoT using machine learning. Multimedia Tools and Applications, 2022, 81(25): 36751-36780.

[12] ElRahman S A, Alluhaidan A S. Blockchain technology and IoT-edge framework for sharing healthcare services. Soft Computing, 2021, 25(21): 13753-13777.

[13] Jiang J, Hua S, Han G, Li A, Lin C. Controversy-adjudication-based trust management mechanism in the internet of underwater things. IEEE Internet of Things Journal, 2022, 10(3): 2603-2614.

[14] Oh J, Park J, Kim Y, Kim K. Algorithm based on Byzantine agreement among decentralized agents (BADA). ETRI Journal, 2020, 42(6): 872-885.

[15] Chatterjee S, Kar A K, Mustafa S Z. Securing IoT devices in smart cities of India: from ethical and enterprise information system management perspective. Enterprise Information Systems, 2021, 15(4): 585-615.

[16] Jiang X, Sun A, Sun Y, Luo H, Guizani M. A trust-based hierarchical consensus mechanism for consortium blockchain in smart grid. Tsinghua Science and Technology, 2022, 28(1): 69-81.

[17] Min Y A, Lim D K. Performance Analysis of Consensus Algorithm considering NFT Transaction Stability. The Journal of the Institute of Internet, Broadcasting and Communication, 2022, 22(2): 151-157.

[18] Fu W, Wei X, Tong S. An improved blockchain consensus algorithm based on raft. Arabian Journal for Science and Engineering, 2021, 46(9): 8137-8149.

[19] Rong B, Zheng Z. FRCR: Raft consensus scheme based on semi asynchronous federal reconstruction. IEEE Transactions on Network and Service Management, 2022, 19(4): 3822-3834.

[20] Wang L, Bai Y, Jiang Q, Leung V C, Cai W, Li X. Beh-Raft-Chain: a behavior-based fast blockchain protocol for complex networks. IEEE Transactions on Network Science and Engineering, 2020, 8(2): 1154-1166.

[21] Guo H, Li W, Nejad M. A hierarchical and location-aware consensus protocol for iot-blockchain applications. IEEE Transactions on Network and Service Management, 2022, 19(3): 2972-2986.

[22] Li D, Zhang F, Tian Y. Research on enterprise management integration mechanism and information platform by internet of things. Journal of Intelligent & Fuzzy Systems, 2020, 38(1): 163-173.

[23] Miloslavskaya N, Tolstoy A. IoTBlockSIEM for information security incident management in the internet of things ecosystem. Cluster Computing, 2020, 23(3): 1911-1925.

[24] Shahjalal M, Islam M M, Alam M M, Jang Y M. Implementation of a secure LoRaWAN system for industrial Internet of Things integrated with IPFS and blockchain. IEEE Systems Journal, 2022, 16(4): 5455-5464.

[25] Showkat S, Qureshi S. Securing the Internet of Things Through Blockchain Approach: Security Architectures, Consensus Algorithms, Enabling Technologies, Open Issues, and Research Directions. International Journal of Computing and Digital Systems, 2023, 13(1): 97-129.