2022 IEEE the 14th International Conference on Wireless Communications and Signal Processing

Location-Based Reliable Sharding in Blockchain-Enabled Fog Computing Networks

1st Xiaoge Huang
School of Communication and Information Engineering
Chongqing University of Posts and Telecommunications
Chongqing, China
huangxg@cqupt.edu.cn

2nd Hongbo Yin
School of Communication and Information Engineering
Chongqing University of Posts and Telecommunications
Chongqing, China
s210101168@stu.cqupt.edu.cn

3rd Yongsheng Wang
School of Communication and Information Engineering
Chongqing University of Posts and Telecommunications
Chongqing, China
s190131050168@stu.cqupt.edu.cn

4th Qianbin Chen
School of Communication and Information Engineering
Chongqing University of Posts and Telecommunications
Chongqing, China
chenqb@cqupt.edu.cn

5th Jie Zhang
School of Communication and Information Engineering
University of Sheffield
United Kingdom
jie.zhang@sheffield.ac.uk

2022 14th International Conference on Wireless Communications and Signal Processing (WCSP) | 978-1-6654-5085-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCSP55476.2022.10039136

***Abstract*—With the explosive growth of the internet of things (IoT) devices, there are amount of data requirements and computing tasks. Fog computing network that could provide computing, caching and communication resources closer to IoT devices (ID) is considered as a potential solution to deal with the vast computing tasks. To improve the performance of the fog computing network while ensuring data security, blockchain technology is enabled and a location-based reliable sharding (LRS) algorithm is proposed, which jointly considers the optimal number of shards, the geographical location of fog nodes (FNs), and the number of nodes in each shard. Firstly, the reliable sharding result is based on the reputation values of FNs, which are related to the decision information and historical reputation value of FNs in the consensus process. Moreover, a reputation based PBFT consensus algorithm is adopted to accelerate the consensus process. Furthermore, the normalized entropy is used to estimate the proportion of malicious nodes and optimize the number of shards. Finally, simulation results show the effectiveness of the proposed scheme.**

***Index Terms*—Blockchain, sharding, fog computing network, internet of things**

I. Introduction

With the rapid growth of the internet of things (IoT) device, the expansion of IoT applications will create amount of computing tasks, resulting in decreasing of the Quality of Service (QoS) in the network. Fog computing networks could provide closer computing, caching and communication resources to IoT devices (IDs) to reduce transmission latency and release the stress on the cloud. Generally, Fog nodes (FNs) are randomly distributed in an unsupervised environment, which leads to user privacy and data security issues during the task offloading process.

Blockchain technology, as the core technology behind Bitcoin, has been gained widespread public attention in the academic and industrial circle, due to it's security and immutability [1], [2]. It could be applied in fog computing networks to ensure the user data security by relevant technologies, such as hashing algorithm, asymmetric encryption, digital signature, etc. Practical Byzantine Fault Tolerance (PBFT) algorithm is a traditional consensus algorithm in blockchain, which could provide $1/3$ Byzantine fault-tolerant capability. However, due to its scalability limitations, defined as the number of processed transactions per second, the PBFT algorithm can not be directly used in fog computing networks.

In [3], the authors proposed a blockchain-assistant fog computing network, which adopted a resource authentication mechanism based on Proof of Work (PoW). A reputation based Byzantine fault tolerance (RBFT) algorithm was proposed in [4], which combines the reputation model to evaluate the decision of each node in the consensus process. In [5], the authors introduced a blockchain-based FN clusters (FNCs) scheme to ensure the network security. However, scalability is an important performance indicator in blockchain networks [6]–[8], and the network performance will be limited by scalability limitations of blockchain.

In this paper, in order to improve the performance of the fog computing network while ensuring data security, a blockchain-enabled fog computing network is considered. A

This work is supported by the National Natural Science Foundation of China (NSFC) (61831002), and Innovation Project of the Common KeyTechnology of Chongqing Science and Technology Industry (Grant no.cstc2018jcyjAX0383).

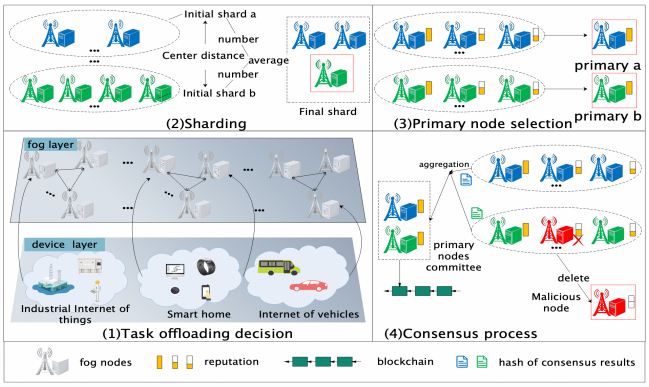


Fig. 1. Blockchain-enabled fog computing networks.

location-based reliable sharding (LRS) algorithm is proposed to improve the throughput of the blockchain, which jointly considers the optimal number of shards, the geographical location of FNs, and the number of nodes in each shard. The reliable sharding result is based on the reputation values of FNs, which are related to the decision information and historical reputation value of FNs in the consensus process. Moreover, a reputation based PBFT consensus algorithm is adopted to accelerate the consensus process. Furthermore, the normalized entropy is used to estimate the proportion of malicious nodes and optimize the number of shards. Finally, simulation results demonstrate that the proposed algorithm can obtain a considerable performance improvement in throughput and latency by consensus process.

The rest of this paper is organized as follows. Section II presents the system model, which contains the network model, the reputation model, and the consensus model. In section III, the LRS algorithm is proposed to optimize blockchain throughput and improve network performance while ensuring network security. Simulation results are presented and discussed in section IV. Finally, section V concludes the paper.

II. System Model

A. Network Model

Consider a blockchain-enabled fog computing network, which includes X IDs and N FNs. FNs could provide computing, caching, communication resource for IDs, while responsible for block generation and verification simultaneously. The latency-sensitive and computation-intensive tasks generated by IDs could be offloaded to neighboring FNs. The reputation values of FNs could be calculated by the decisions in the consensus process and the historical reputation of FNs. In addition, the reputation values of FNs are recorded in the blockchain to select the primary node of each shard and identify the potential malicious nodes. As shown in Fig. 1, there are four steps for offloading and recording of IDs tasks, namely, task offloading decisions, sharding, primary node selection and consensus process. The details of each step are summarized as follows.

1) Task offloading decision: Firstly, task offloading decisions are made by the QoS-aware resource allocation algorithm proposed in [9]. Secondly, establish the connection between IDs and associates, and offload the computing tasks to FNs. Finally, the task offloading process information will be recorded in the blockchain by FNs.

2) Sharding: Firstly, the proportion of malicious nodes in the network is calculated by the normalized entropy method. Secondly, the optimal number of shards k is calculated according to the proportion of malicious nodes. Then, FNs are divided into k shards. Finally, dynamic adjust the FNs in the shard to ensure all FNs evenly into each shard by LRS algorithm.

3) Primary node election: Reputation based PBFT consensus algorithm is adopted in shard. Selected each primary node in each shard. FN with a higher reputation value, the higher the probability of being selected as the primary node.

4) Consensus process: Each shard constructs a sub-chain and the primary node in each shard is responsible for packaging all unconfirmed transactions and broadcasting them in the shard. Remaining FNs in the shard are replicas to verify the broadcast transactions. After the consensus of each shard, the primary nodes of each shard will add the blocks to the sub-chain and upload the hash summary and the decision information to the Primary Node Committee (PNC) for aggregation, which is composed of the primary nodes of each shard. Finally, the PNC will add the aggregated information and FNs reputation values into the main-chain, and then identify malicious nodes according to the updated reputation value.

B. Reputation Model

Reputation value indicates the trust degree of a FN. In the consensus process, FNs could have following three kinds of decisions: the correct decision, when FNs successfully participate in the block generation process; the wrong decision, when FNs fail to generate a block, or disagree with the majority; the abstention decision, when FNs did not participate in the consensus process. After the consensus process, the number of each kind of decisions could be obtained. The reputation value of FN n, $\theta_n \in [-1, 1]$, will be updated based on the following formulas.

$$\theta_n = \begin{cases} \frac{\theta_1 a - \theta_2 b - \theta_3 c}{a+b+c}, \text{correct decision;} \\ \frac{-\theta_1 a + \theta_2 b - \theta_3 c}{a+b+c}, \text{wrong decision;} \\ \frac{-\theta_1 a - \theta_2 b + \theta_3 c}{a+b+c}, \text{abstention decision;} \end{cases} \tag{1}$$

where a, b and c represent the number of correct, wrong and abstention decisions, and the associated weights are θ_1, θ_2 and θ_3 respectively.

$$\theta_1 = \frac{F(a)}{F(a) + F(b) + F(c)} \tag{2}$$

$$\theta_2 = \frac{F(b)}{F(a) + F(b) + F(c)} \tag{3}$$

$$\theta_3 = \frac{F(c)}{F(a) + F(b) + F(c)} \tag{4}$$

where, $F(\cdot)$ controls the scores sensitivity.

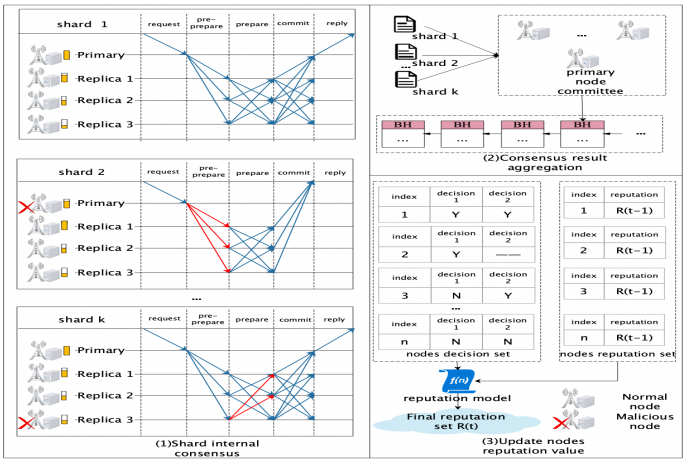


Fig. 2. Reputation-based PBFT consensus process in blockchain-enabled fog computing network.

Let $R_n(t-1)$ denote the reputation value of FN n at time slot $t-1$, thus $R_n(t) = R_n(t-1)+\theta_n$ is the reputation value of FN n at time slot t.

If the reputation value $R_n(t)$ of FN n is lower than the given threshold, then FN n is considered as the malicious node or the node with poor performance. Therefore, FN n will be removed by the network administrator, until it is repaired.

C. Consensus Model

As shown in Fig. 2, in the blockchain-enabled fog computing network FNs are divided into k shards, each shard processes and verifies transactions parallelly. The reputation based PBFT algorithm is adopted in each shard. The primary node is responsible for packaging and broadcasting all unconfirmed transactions in the shard, named the pre-preparation phase.

Replicas receive the pre-preparation information, verify the authenticity of the data of the block, and send their verification decisions to other replicas, named the preparation phase. When a replica receives more than $\frac{2}{3}$ of the total number of preparation messages, it could proceed to the next phase.

If most of the preparation messages are received by the replicas indicates that the primary node has packaged the block correctly. Then it will broadcasts its decision to other replicas in the shard, named the commit phase. Otherwise, the primary node will be changed. Finally, when replicas receive more than $\frac{2}{3}$ of the commit messages, which indicates that the block can be recorded and the internal shard consensus is successful. Then, the primary node will add the block to the sub-chain, upload the hash summary of the consensus result and the decision information to the PNC.

The PNC consists of the primary nodes of each shard, which is responsible for aggregating the hash summary of consensus results in each shard and calculating the reputation value of FNs. After receiving the consensus results and decisions of all shards, PNC will aggregate them and update the reputation values of FNs at the same time. Then PNC will record the reputation values of FNs into the main-chain and publish it to the network. In addition, if the primary node in a shard is changed, the previous primary node is considered as a malicious node, and its reputation value will be set to 0. If the reputation value of a FN is lower than the given reputation threshold, the node is considered as a malicious node or a node with poor performance.

III. Location-based Reliable Sharding Optimization Algorithm

A. Proportion of malicious nodes

Honesty FNs will tend to promote correct consensus and resist malicious consensus, but they may fail to participate in the consensus due to a lack of computing resources or external factors. Based on behaviors of FNs, FNs are divided into three categories, namely, $\zeta_n = \{1, 0, -1\}$, which indicate that FN n made correct decisions, wrong decisions and abstention (did not participate in the decision of this round), respectively.

Moreover, malicious FNs tend to promote the malicious consensus and resist the correct consensus, which will make wrong or abstention decisions. Based on behaviors of malicious FN n', FN n' are divided into two categories, namely, $\zeta_{n'} = \{0, -1\}$, which refers to wrong or abstention decision, respectively.

After the consensus process in the shard, the primary node will aggregate the decision information of FNs in the preparation phase and the commit phase. For FN n, the validation decision in the preparation phase or the execution decision in the commit phase can be expressed as:

$$\vartheta_n = \begin{cases} 1, \text{correct decision;} \\ 0, \text{wrong decision;} \\ -1, \text{abstention decision;} \end{cases} \tag{5}$$

Based on decisions of FNs in shard i, the decision matrix at the preparation phase, the execution matrix at the commit phase, pre_i, com_i, are given respectively:

$$pre_i = [\vartheta_1, \vartheta_2, ..., \vartheta_{sum}], com_i = [\vartheta_1, \vartheta_2, ..., \vartheta_{sum}] \tag{6}$$

The FNs that make correct decisions in both the preparation phase and the commit phase are regarded as honesty nodes. While the FNs that make wrong decisions or did not participate in the decision-making phase, are regarded as malicious nodes. The number of two types of nodes in shard i is recorded as α_i and β_i respectively. Therefore, the number of honesty nodes A and the number of malicious nodes B in the network can be expressed as:

$$A = \sum_{i=1}^{k} \alpha_i, B = \sum_{i=1}^{k} \beta_i \tag{7}$$

Let $\phi_i = \frac{A}{N_i}$ and $\psi_i = \frac{B}{N_i}$ be the total proportion of the correct consensus and the wrong consensus in shard i, respectively, where N_i represents the number of FNs in shard i. Normalized entropy are used to represent inconsistent parameters in the network, which are defined as follows:

$$H = \frac{1}{k}(\sum_{i=1}^{k} -\phi_i \log_2(\phi_i) - \psi_i \log_2(\psi_i)) \tag{8}$$

Based on the proportion of consensus decisions of each shard, the inconsistencies of each shard are similar [10]. Assume the proportion of malicious nodes in the network is p, which can be calculated by normalized entropy:

$$H = -\overline{p}\log_2(\overline{p}) - (1-\overline{p})\log_2((1-\overline{p})) \tag{9}$$

$$p = \min(\overline{p}, 1-\overline{p}) \tag{10}$$

B. Optimal number of shards

We assume the worst-case scenario, where all malicious nodes are assigned to one shard. Therefore, as long as this shard can work properly, the entire network can also work properly. Assume the number of shard is k and the proportion of malicious nodes p could be obtained by (10). The fault tolerance of the traditional PBFT algorithm is $\frac{1}{3}$. Based on this, assume the number of honesty nodes is N_h, and the number of malicious nodes is N_m, the sharding process includes the following two cases [10].

Cases 1: The primary node is an honesty node, the number of honesty nodes in the decision-making process is $N_h - 1$. To ensure the consensus properly, the following inequality should be satisfied: $N_h - 1 \geq 2N_m$.

where, the sum of the number of honesty nodes and malicious nodes is equal to the total number of nodes in the shard, that is, $N_h + N_m = N/k$, while N represents the number of FNs. The number of malicious nodes is calculated by $N_m = N \times p$. Thus, the number of shard in case 1 can be calculated by $k \leq \frac{N}{3N\times p+1}$.

Cases 2: The primary node is a malicious node, so the number of honesty nodes in the decision-making process is N_h, and the number of malicious nodes is $N_m - 1$. To ensure that the shard works properly, the following inequality should be satisfied: $N_h \geq 2N_m - 1$.

Simultaneously, the number of shards in case 2 can be calculated by $k \leq \frac{N}{3N\times p-1}$. Combining the above two cases, the optimal number of shards to ensure network security can be obtained:

$$k = \frac{N}{3N \times p + 1} \tag{11}$$

C. Location-based reliable sharding algorithm

We proposes the LRS algorithm to maximize blockchain throughput while ensuring network security, which includes following steps:

Step 1: The blockchain-enabled fog computing network is divided into k shard by K-Means clustering algorithm. Firstly, select k center points randomly in the coverage of FNs, denoted as $cen = (cen_1, cen_2, ..., cen_k)$. Secondly, calculate the Euclidean distance between FN n and k centers, and define the objective function $O = \min \sum_{i=1}^{k} \sum_{n=1}^{N} \|FN_n^i - cen_i\|^2$. Finally, update the center until the objective function converges to obtain the initial node set of shards.

Step 2: Calculate the number of nodes v in each shard, where $v = N/k$. Then, adjust FNs based on their geographical locations, so that they are equally distributed in each shard. Therefore, the number of FNs in each shard is around v. The details of the LRS algorithm is shown in Algorithm 1.

Algorithm 1 LRS Algorithm

1: Initialize: k, N and FNs information matrix $FN = \{FN_1, FN_2, ..., FN_N\}$, include the geographical location and other information of FNs
2: Randomly select k initial center points $cen = (cen_1, cen_2, ..., cen_k)$
3: Update the center until the objective function $O = \min \sum_{i=1}^{k} \sum_{n=1}^{N} \|FN_n^i - cen_i\|^2$ converges
4: Calculate $v = N/k$
5: Adjust FNs in each shard, until FNs are equally distributed into k shards

IV. Simulation Results and Analysis

A. Simulation scenario and parameter setting

In this section, we verify the effectiveness of the proposed algorithms by MATLAB software simulation. Consider a blockchain-based fog computing network consisting of N FNs and X IDs, which are randomly distributed over a $1500m \times 1500m$ region. In the simulation, the channel model $L(d(n,n\prime)) = 140.7 + 36.7 * \log_{10} d(n,n\prime)$ based on 3GPP LTE-Advance outdoor scene is used, where $d(n,n\prime)$ represents the distance between FN n and FN $n\prime$. The number of FNs and malicious FNs is 40 and 6.

B. Simulation results

Fig. 3 shows the performance of the LRS algorithm in blockchain-enabled fog computing networks proposed in this paper. It can be seen from the result, the proposed LRS algorithm can effectively identify and reduce the proportion of malicious nodes. In addition, with the increases number of shards the throughput of the blockchain is also increased. As shown in fig. 3, with the increase of iterations, malicious nodes in the network can be identified and removed. Therefore, after several iterations, the proportion of honesty nodes gradually increases, while the proportion of malicious nodes gradually decreases. After 8 iterations, the proportion of malicious nodes decreases from 0.325 to 0.025. As the proportion of malicious nodes decreases, the number of shards increases from 1 to 10.

Fig. 4 shows the results of node division at two steps of the LRS algorithm proposed in this paper. In the first step, FNs are divided into k shard by K-Means clustering algorithm. In the second step, FNs are evenly distributed to each shard by adjusting the number of FNs of each shard. Fig. 4 (a) shows the results of the LRS algorithm in the first step. The triangle symbol is used to represent FNs, where colors are used to distinguish the shard. Taking the data of the iteration 6 in Fig. 3 as reference (the number of shard is 3), the proposed LRS algorithm is used to initially divide FNs into three shards. Fig. 4 (b) shows the results of the LRS algorithm in the second step. By adjusting the FNs of each shard, FNs are evenly distributed into each shard. On the one hand, it avoids too many nodes

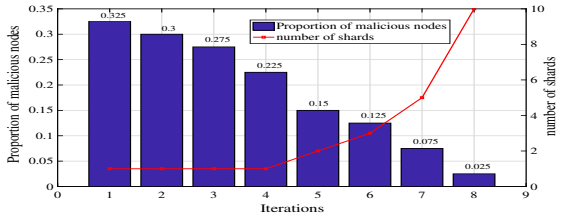
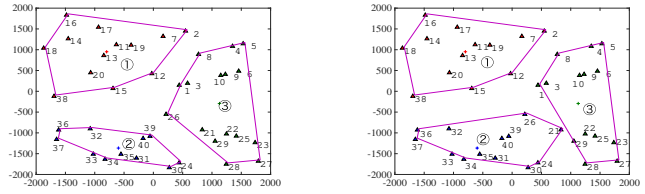
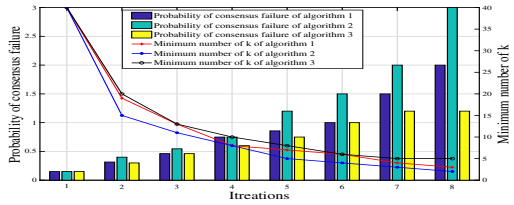


Fig. 3. Proportion of malicious nodes and the number of shards versus iterations.

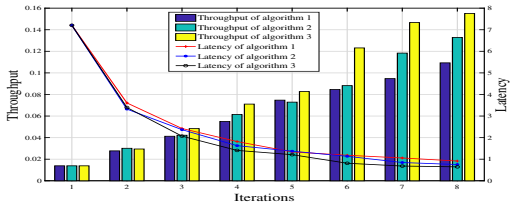


(a) (b)

Fig. 4. Initial and final sharding results by the LRS algorithm.



(a)



(b)

Fig. 5. Performance comparison of three algorithms. (a) Probability of consensus failure and minimum number of k versus iterations. (b) Network throughput and latency versus iterations.

assigned to one shard, resulting in high consensus latency. On the other hand, it also avoids too few nodes allocated to one shard, resulting in low security. It can be seen from the Fig. 4 (b), FN 26 and FN 21 are adjusted from shard 3 to shard 2.

In Fig. 5, we show the changes of probability of consensus failure, minimum number of k, network throughput and latency for different algorithms versus itraetions. Consider three comparison algorithms [10]. Algorithm 1 is based on the reputation value of FNs for sharding, and algorithm 2 is based on the geographical location of FNs for sharding, which are only consider single parameter. Algorithm 3 is the LRS algorithm proposed in this paper.

Algorithm 1 assigns FNs without considering the geographic location of FNs, which increases the communication latency and reduces the throughput. Algorithm 2 only considers the geographic location of FNs resulting in few FNs assigned to one shard and a high probability of consensus failure. As shown in Fig. 5 (a) and (b), the performance of the LRS algorithm proposed in this paper is greatly improved in all aspects compared with the comparison algorithms.

V. Conclusion

To guarantee the security of the fog computing network, the blockchain-enabled fog computing network model based on sharding technology was proposed. The fog computing network was divided into multiple parallel shards by sharding technology, which reduces the blockchain consensus latency, thus improving the throughput of the blockchain in the fog computing network. The reputation of FNs was updated to identify malicious nodes in the network. The normalized drop method was used to calculate the proportion of the number of malicious nodes in the network and calculate the optimal number of shards. Then, the LRS algorithm was proposed, which jointly considered the optimal number of shards, the geographical location of FN, and the number of nodes in each shard. This algorithm could optimize blockchain throughput and improve network performance while ensuring network security. Finally, simulation results demonstrated the effectiveness of the proposed scheme.

References

[1] J. Gao et al., "A Blockchain-SDN-Enabled Internet of Vehicles Environment for Fog Computing and 5G Networks," in IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4278-4291, May 2020.

[2] L. Yang, M. Li, H. Zhang, H. Ji, M. Xiao and X. Li, "Distributed Resource Management for Blockchain in Fog-Enabled IoT Networks," in IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2330-2341, 15 Feb.15, 2021.

[3] T. Chen, L. Zhang, K. -K. R. Choo, R. Zhang and X. Meng, "Blockchain-Based Key Management Scheme in Fog-Enabled IoT Systems," in IEEE Internet of Things Journal, vol. 8, no. 13, pp. 10766-10778, 1 July1, 2021.

[4] K. Lei, Q. Zhang, L. Xu and Z. Qi, "Reputation-Based Byzantine Fault-Tolerance for Consortium Blockchain," 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), 2018, pp. 604-611.

[5] D. Wu and N. Ansari, "A Cooperative Computing Strategy for Blockchain-Secured Fog Computing," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6603-6609, July 2020.

[6] D. Liu, A. Alahmadi, J. Ni, X. Lin and X. Shen, "Anonymous Reputation System for IIoT-Enabled Retail Marketing Atop PoS Blockchain," in IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3527-3537, June 2019.

[7] G. Xu, Y. Liu and P. W. Khan, "Improvement of the DPoS Consensus Mechanism in Blockchain Based on Vague Sets," in IEEE Transactions on Industrial Informatics, vol. 16, no. 6, pp. 4252-4259, June 2020.

[8] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang and R. P. Liu, "Survey: Sharding in Blockchains," in IEEE Access, vol. 8, pp. 14155-14181, 2020.

[9] X. Huang, Y. Cui, Q. Chen and J. Zhang, "Joint Task Offloading and QoS-Aware Resource Allocation in Fog-Enabled Internet-of-Things Networks," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7194-7206, Aug. 2020.

[10] X. Huang, Y. Wang, Q. Chen and J. Zhang, "Security Analyze with Malicious Nodes in Sharding Blockchain Based Fog Computing Networks," 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 2021, pp. 1-5.