Information Sciences 561 (2021) 70–80

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Blockchain-based mobile edge computing system

Guangshun Li [a], Xinrong Ren [a], Junhua Wu [a,*], Wanting Ji [b], Haili Yu [a], Jiabin Cao [a], Ruili Wang [b]

[a] School of Computer Science, Qufu Normal University, Shandong, China
[b] School of Natural and Computational Sciences, Massey University, Auckland, New Zealand

ARTICLE INFO

Article history:
Received 13 August 2020
Received in revised form 18 December 2020
Accepted 17 January 2021
Available online 2 February 2021

Keywords:
Edge computing
Blockchain
Clone blocks
Neural networks
Sharing

ABSTRACT

With the development of the Internet of Things (IoT), the number of mobile terminal devices is increasing rapidly. Due to high transmission delay and bandwidth limitation, computing power requirements for IoT devices are getting higher and higher. Recently, edge computing is an effective way to reduce system delay, and blockchain solves the security problem of edge computing. In this paper, a three-layer network model, named blockchain-based mobile edge computing system (BMEC), is proposed for clone block identification. Specifically, a neural network based clone block identification (NCBI) method is proposed to prevent clone block attacks. After that, the Prim algorithm is applied to BMEC to generate a weighted undirected graph minimum spanning tree that is composed of edge blocks. This can divide a main chain into several side chains to improve the transaction speed of blockchain. Finally, the blockchain is constructed based on the time slicing round-robin scheduling algorithm to control resources from edge servers and regulate edge devices' activities based on the predefined rules of priority, application type, and past behavior. Experimental results show that our clone block identification method can achieve block validation effectively in BMEC, and our construction of blockchain delay is lower than conventional edge computing methods.

1. Introduction

The Internet of Things (IoT) devices collect data and deliver the data to users, who use the data for analysis and decision making. However, in the context of the growing popularity of the IoT, some applications need lower latency, such as self-driving cars, and it is becoming more and more important to shift computing power closer to the edge of the network to reduce costs [18]. Edge computing is based on the services and applications of the edge server between the user layer and the cloud data center. It is a distributed platform integrating network, computing, storage and application processing capabilities. It provides data resource services for users at the edge of the network near the IoT devices or data sources [32].

When a cloud server storage and computing burden is too heavy, and the network transmission bandwidth pressure is too heavy, edge computing, as a medium of IoT devices and cloud computing, becomes useful and necessary, since edge computing can process local data faster than cloud computing in near real-time. Thus, edge computing can effectively reduce system delay [25]. Edge computing extends cloud computing service resources to the edge of the network, and solves the problems of poor cloud computing mobility, weak geographic information awareness, and high cloud computing latency

* Corresponding author.
E-mail address: shdwjh@163.com (J. Wu).

https://doi.org/10.1016/j.ins.2021.01.050
0020-0255/

[34]. At the same time, edge computing distributes computing tasks to multiple nodes, and each node has the ability to make independent judgments and decisions [28]. When one node is destroyed, other nodes will not be greatly affected, thereby improving the system's ability to resist risks [19].

Blockchain is a distributed architecture with the characteristics of decentralization, transparency and openness, and cannot be tampered with. It is also the basic technology of Bitcoin. Usually, string data can be added to a blockchain through encryption [12]. Under the trust mechanism based on credit identification, authoritative third-party organizations (such as banks) support traditional social trust [11]. Therefore, without a third-party central node, it is difficult to directly establish trust between two strange entities.

Blockchain can solve the problem of trust establishment among distributed system through distributed node verification and consensus mechanisms [27]. Specifically, the distributed architecture of a blockchain is consistent with the distributed architecture of the IoT [31]. Building a blockchain into the IoT can complete value transfer while transmitting information. The architecture of blockchain has realized a major transformation from "Information Internet" to "Value Internet" [4].

In this paper, we integrate the concept of blockchain used in cryptocurrency into our three-tier system architecture (including a IoT device layer, an edge server layer, and a cloud computing network layer) for edge computing. After successful implementation, the proposed system can make full use of the computing and storage resources of all participants, thereby improving system efficiency and scalability. However, due to the slow transaction speed in the blockchain, there is a large delay in the edge computing system based on the blockchain, and the block is connected to the node, making the nodes in the BMEC face a series of security challenges, such as clone block attack.

In order to reduce the delay of BMEC and ensure the security of blockchain nodes, this paper will solve the following three problems through the proposed three-tier network architecture:

(i) In the IoT device layer, since IoT devices are in danger of being cloned when they are stored in blocks, how to identify clone blocks is the first problem to be solved.
(ii) In the edge server layer, how to divide the blockchain network of this layer to improve the transaction speed is the second problem to be solved.
(iii) In the cloud computing network layer, since this layer needs to process a large amount of block data, how to achieve load balancing and reduce the generation delay of the blockchain is the last problem to be solved.

This paper aims to (i) optimize the construction method of blockchain at the central level of cloud computing networks, (ii) topologize the blockchain network at the edge server layer, and (iii) resist clone block attacks at the IoT device layer. To achieve this, this paper proposes a mobile edge computing system based on blockchain. Specifically, a neural network based clone block identification (NCBI) method is proposed to identify clone blocks. For multiple blockchain generation, we use the time slicing round-robin scheduling algorithm [23] to determine the next generated block, and allocate multiple block tasks to different resource pools to achieve efficient construction of the blockchain. The main contributions are summarized as follows:

(i) In the IoT device layer, the NCBI method is proposed to identify whether the unknown block is a clone block.
(ii) In the edge server layer, we use the Prim algorithm [7] to divide the blockchain network. It improves the transaction throughput of the blockchain.
(iii) In the cloud computing layer, we use the time slicing round-robin scheduling algorithm [23] to generate the blockchain.

The rest of this paper is organized as follows. We discuss the related work in Section 2. We describe the proposed system model in Section 3. In Section 4, we provide a safe and efficient way to build a mobile edge computing system based on blockchain. Section 5 is our experimental simulation results and analysis. Finally, the conclusion is summarized in Section 6.

2. Related work

The performance bottleneck of the limited resources of the blockchain makes it difficult to apply blockchain technology to large-scale IoT terminal devices. Edge computing provides a convenient, low-latency, and distributed computing offloading platform for mobile devices with limited resources. Some work combined blockchain and IoT to solve this problem.

To build a security architecture system for the edge blockchain, Xiong et al. [24] proposed a prototype of mobile edge computing enabled blockchain systems. Gai et al. [9] proposed a blockchain-edge scheme that utilizes dynamic programming to produce optimal solutions to selecting global transaction paths. Ma et al. [13] proposed a configurable scheme blockchain architecture protocol based on BlockTDM (Blockchain-based trusted data management scheme), in which a flexible and configurable blockchain architecture was proposed, including mutual authentication protocols, flexible consensus and smart contract, block and transaction data management, blockchain nodes management and deployment. Pan et al. [14] defined edge chains and proposed an edge IoT system based on blockchain and smart contract. To improve the speed of data processing in edge devices, Zhang et al. [30] proposed a k*Tree classification method to accelerate the test stage and improve the speed of data processing by storing the information of the training sample in the leaf node of a k-tree. However, these works were based on edge computing systems, while neglected the security of data block itself.

Later, hash functions were introduced for ensuring the security of blockchains. Qi et al. [15] extended the traditional Locality-Sensitive Hashing (LSH) function to incorporate the time factor. Then they also proposed a time-aware and privacy-preserving service recommendation approach based on LSH. Wang et al. [22] proposed a method that can diversify recommendation lists named DivRec_LSH based on historical usage records and LSH. Zhong et al. [33] proposed a multi-dimensional quality-driven service recommendation method with privacy protection in the mobile edge environment. These works were optimized on hash functions to ensure the reliability of the blockchain itself. In addition, Song [16] proposed a detection algorithm for malicious nodes based on reliable metrics (RDICS) to identify malicious nodes and reduced the fusion weights of these malicious nodes, thus weakening their contribution to the final decision. However, when the end-users store data to the blocks, the data blocks may receive the attack of clone blocks, resulting in block data disclosure.

Recently, artificial neural networks became a research hotspot, which provided a new idea to study the identification of clone blocks. Wang et al. [21,20] proposed stochastic configuration algorithms, and generated a stochastic configuration network learning model to integrate heterogeneous features for large-scale data analysis. Zhang et al. [29] proposed a neural network model, which extended the dimension of connection weights from one to multiple. Wang et al. [5] proposed a survey on an emerging area: deep learning for smart city data. The above work demonstrated that artificial neural networks have been widely used in big data processing and IoT applications. Inspired by the above, this paper aims to explore the use of artificial neural networks with blockchain to ensure block data security in the process of identifying clone blocks.

In recent years, the application of blockchain for IoT is another research hotspot. Gai et al. [10] proposed an edge internet (BloE) model based on blockchain, which used the characteristics of blockchain to solve the problem of task allocation in edge computing, and prevented blocks from being attacked maliciously by differential privacy technology. Fu et al. [8] proposed an optimal power and resource allocation scheme to meet the demands of the computational costs for blockchain-based IoT devices. Sharma et al. [17] proposed an edge node scheme using a multi-layer blockchain, applying edge block technology as a carrier the allocation of resources to use devices. Xu et al. [26] proposed a two-stage offloading optimization strategy for jointly optimizing offloading utilities and privacy in edge computing enabled IoT for edge computing resource offloading. Pietro et al. [6] considered and analyzed a new resource scheduling scheme for edge blocks implemented by blockchain network nodes, which periodically updated and aggregated blockchain data and further reduced the communication costs of connected IoT devices.

To solve the block joining problem of blockchain systems, the hyperledger fabric [2] in the prior art adopted a multi-party permission method. When most of blocks in the blockchain system are reviewed and approved, the requested block was added to the block Chain's permission list. The proof-of-authority method [3] used a multi-party request method. After most nodes in the blockchain system had submitted a request to join, the requesting node was added to the permission list of the blockchain. The above two works used community-based methods to verify from the administrators of other nodes. The addition of the block was not completed until the majority of people approved the request block. However, these methods ignored the waiting time of multiple other request blocks in the queue.

To solve the above problem, this paper proposes a three-tier system, named blockchain-based mobile edge computing (BMEC) system, which contains an IoT device layer, an edge server layer, and a cloud computing network layer. In BMEC, clone blocks can be identified efficiently in the IoT device layer, and the blockchain transaction is realized in the edge server layer. Thus, BMEC reduces the technical threshold and cost of the system. Further, we propose a neural network based clone block identification (NCBI) method to prevent clone block attacks. Then we applied the Prim algorithm [7] to generate a weighted undirected graph minimum spanning tree that is composed of edge blocks. In addition, the time slicing round-robin scheduling algorithm [23] is used to ensure that each requested block in the queue can be executed and authenticated fairly within a limited time.

3. The proposed BMEC system

In this section, we discuss some key considerations for BMEC design. Fig. 1 shows the overall architecture of BMEC, which consists of mobile terminal devices, edge servers, cloud servers, and blockchain. Specifically, each mobile terminal device in the IoT device layer is constructed as a block with smart contracts. BMEC creates a blockchain account and manages these devices from the background. The interconnected edge servers in the edge network layer are defined as specific blockchains on the edge layer and are responsible for monitoring the transactions in BMEC. New blocks will be added when blockchain transactions. The interconnected cloud servers in the cloud layer are considered as a blockchain network on the cloud. By optimizing the topology of the blockchain network, the proposed system can increase the transaction speed of the cloud computing layer blockchain.

3.1. Clone block

A clone block attack means that an illegal attacker in a blockchain captures a legitimate block and acquires its legal identity information, copies out several blocks with their block ID and key verification information, and drops these clone blocks to different locations in the network to launch the attack. Because the clone blocks have the same ID and key information as a legitimate block, the traditional cryptography-based authentication mechanism cannot detect the clone block. Fig. 2 is a schematic diagram of clone blocks attack. When the edge server stores data in a legal block, it may be attacked by multiple

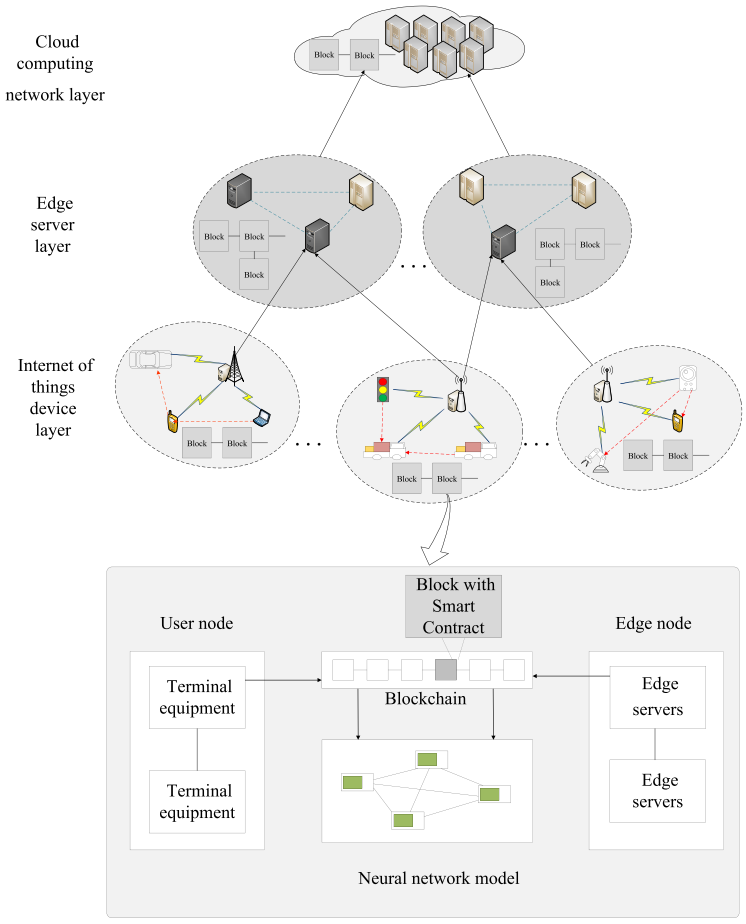


Fig. 1. Blockchain-based mobile edge computing system.

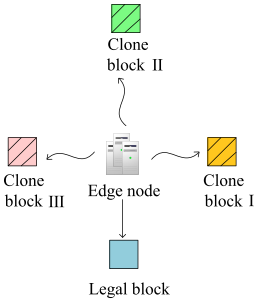


Fig. 2. Schematic diagram of clone blocks attack.

clone blocks. As shown in Fig. 2, block I, II, and III are all clone blocks. The blue block represents the legal block, the yellow block I represents the clone block with false information content greater than 66%, the green block II represents the clone block with false information content between 33% and 66%, and the pink block III represents a clone block with the false information content of less than 33%. In other words, the order of false information content in clone blocks I, II, and III is: clone block I $>$ clone block II $>$ clone block III. The neural network is a model of the human brain nervous system. It is a set of interconnected neurons. We build our own neural network by training some data sets, and then use the generated neural network to classify and predict the real data. Fig. 3 is a schematic diagram of the neural network model.

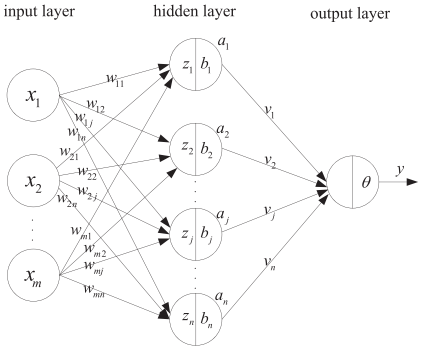


Fig. 3. Neural network model.

3.2. Blockchain network sharing model

The blockchain network at the edge server layer is abstracted as a weighted undirected graph, and using the Prim algorithm [7] to generate the minimum spanning tree. Then we select the subtree with the smallest sum of weights as the main chain, and the other subtree as the side chain. The main chain stores the transaction directory of the blockchain network, and the transaction calculation details of the side chain are synchronized to each block of the main chain. When adding a new block N_i to the edge blockchain network, we need to find the minimum PoW (Proof of Work). After the new block is added, the communication cost between the original block j and the newly added block N_i is a j-dimensional matrix: $W_{ij} = [w_{i1}, w_{i2}, \ldots, w_{ij}]$. The delay of adding a new block N_i to the original blockchain network is $W(T)$, we can get the minimum delay of the blockchain at the edge server layer as the minimum value in the matrix $W(T) = minW_{ij}$.

3.3. Blockchain construction method

For multiple blockchain construction tasks, in order to solve the problem of high delay in the blockchain construction process, we use the time slicing round-robin scheduling algorithm [23] to determine the order of blockchain construction. When multiple blockchains need to generate new blocks, we execute the time slicing round-robin scheduling algorithm [23] to determine the next block to be added. The time slicing round-robin scheduling algorithm [23] binds a large number of new block tasks to the same scheduler, manager, and trigger. Thus, our method can effectively manage the addition of various delay blocks and periodic blocks in blockchain.

3.4. System complexity analysis

The time complexity of the proposed system consists of three parts, the time complexity of a three-layer neural network, the time complexity of the Prim algorithm [7], and the time complexity of the time slicing round-robin scheduling algorithm [23].

In the three-layer neural network, we assume that the number of neurons in each layer is n_1, n_2 and n_3. When the feedforward calculation of the proposed network is carried out, matrix multiplication is carried out twice, that is, the $n_1 \times n_2$ and $n_2 \times n_3$ times calculation are carried out. Since the number of nodes in the input layer, hidden layer and output layer is given (i.e., n_1, n_2 and n_3), they can be regarded as constants. Thus, the time complexity of feedforward calculation of the proposed network is $O(n_1 \times n_2 + n_2 \times n_3) = O(n^2)$. The back-propagation time complexity is the same as the feedforward calculation. The activation function of the network is ReLU function with low computational complexity and is ignored here. Therefore, the time complexity of the proposed network is $O(n^2)$.

The basic idea of the Prim algorithm [7] is: firstly, a node is taken as the initial node of the minimum spanning tree, and then the edge with the minimum weight is found iteratively and added to the minimum spanning tree. If a loop is generated after joining, skip this edge and select the next node. When all nodes are added to the minimum spanning tree, the minimum spanning tree in the connected graph is created. Thus, in the proposed system, assuming that the number of vertices is n, the time complexity of the Prim algorithm [7] is $O(n^2)$.

In the time slicing round-robin scheduling algorithm [23], based on the principle of first come first served, the system arranges all the ready processes into a ready queue, generates an interrupt every once in a while, and then the system assigns the server to the next team leader process to start a new round of circulation. Thus, in the proposed system, the time complexity of the time slicing round-robin scheduling algorithm [23] is $O(n)$.

As a result, the total time complexity of our proposed system is $O(n^2)$.

4. Algorithm solution

4.1. A neural network based clone block identification method

Our neural network based clone block identification method mainly includes the following steps:

S1. The legitimate block and the edge node perform upper layer authentication. The edge node stores the identity declaration information sent from the legitimate block as follows:

$$R_i = \{ID_i, H_i\}, \tag{1}$$

where ID_i is the identity number of the block, H_i represents the reliability metric used to extract from the block. The reliability metric has the unique characteristics of time and space, and cannot be copied.

S2. The edge server extracts the identity declaration information $R_i = \{ID_i, H_i\}$ of the legitimate block i and the identity declaration information $R_j = \{ID_j, H_j\}$ of the unknown block j.

S3. Compare whether the ID of blocks i and j are consistent. If $ID_i \neq ID_j$, block j is not a clone block; otherwise, block j is a clone block.

S4. If the ID of blocks i and j are consistent, the edge server extracts the reliability metric of block i and block j, uses the test statistics to calculate the correlation degree, and identifies whether it is a clone block.

S5. If block j is a clone block, the reliability metric of legitimate blocks x_j and clone block y_j are combined into a training set

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}. \tag{2}$$

S6. The edge block uses the neural network algorithm [1] to train data set S until a model that meets the identification rate is generated.

S7. The edge block uses the model that meets the requirements to identify the block information of the new unknown block x_k and get the result

$$y_k = \begin{cases} 1, & x_k \in H_i(n) \\ 0, & x_k \in H_j(n) \end{cases} \tag{3}$$

where y_k is the output of node x_k identified by neural network model. If $y_k = 1$, the unknown block x_k is a legitimate block; otherwise, x_k is a illegal block.

4.2. Sharing model based on the Prim algorithm

We use the Prim algorithm [7] to divide the blockchain network topology at the edge server layer to optimize the communication delay of the blockchain.

We initialize tree T as an empty tree, and then add the $n-1$ edge (u, v) to the tree T until the minimum spanning tree with $n-1$ edge is generated. The Prim algorithm [7] is described as Table 1.

As shown in Fig. 4, we abstract an edge server network as an undirected weighted graph, where matrix M denotes the adjacency matrix corresponding to Fig. 4. The process of generating the minimum spanning tree fragment model using the Prim algorithm [7] in a blockchain network is shown in Fig. 5.

$$M = \begin{bmatrix} 0 & 6 & 1 & 5 & \infty & \infty \\ 6 & 0 & 5 & \infty & 3 & \infty \\ 1 & 5 & 0 & 5 & 6 & 4 \\ 5 & \infty & 5 & 0 & \infty & 2 \\ \infty & 3 & 6 & \infty & 0 & 6 \\ \infty & \infty & 4 & 2 & 6 & 0 \end{bmatrix} \tag{4}$$

Table 1
Prim algorithm.

Prim algorithm:
Input: the weighted undirected graph $G = V, E$, where the vertex set is V and the edge set is E 1. $x \in V, V_{new} = \{x\}, E_{new} = \{\}$ 2. for $V_{new} = 1:1:v$ do choose the edge with the smallest weight $< u, v > \in E, u \in V_{new}$, $v \notin V_{new} \bigcap v \in V$ 3. Add v to $V_{new}, < u, v >$ to E_{new}. output: minimum spanning tree $G = \{V_{new}, E_{new}\}$

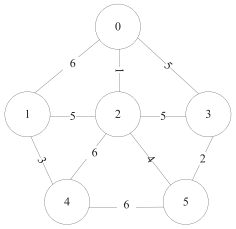


Fig. 4. Undirected weighted graph.

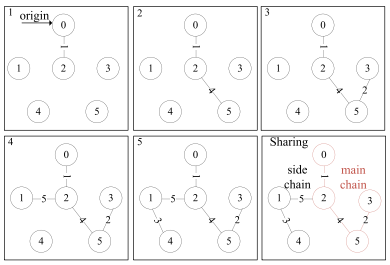


Fig. 5. Sharing model based on the Prim algorithm.

4.3. Blockchain generation method based on time slicing round-robin scheduling

When building a blockchain, we use the time slicing round-robin scheduling algorithm [23] to determine the next generated block.

In the blockchain generation method: the number of blocks requested to be added to the blockchain is n, Q is the queue of n requested blocks, c_i is the ith requested block, t_i is the execution time of block c_i, and t_u is the unit time.

S1. The n request blocks are arranged in the queue Q according to the first-come-first-served order, the edge server selects the block c_i from the queue Q and adds it to the blockchain.
S2. If $t_i \leqslant t_u$, the server will immediately release the memory after c_i execution, and the scheduler will then process the next request block c_{i+1}; otherwise, the server will interrupt the execution of the block c_i, and the block c_i is added to the end of the queue Q, and then the server will select the next block c_{i+1} in the Q queue to execute the service.
S3. Repeat the above two steps until all blocks in the Q queue are added to the blockchain.

The above process can ensure that each block is selected and executed without recording all current connection states, which is a stateless connection and can concisely realize the construction of blockchain.

5. Experimental simulation results analysis

5.1. Experimental data sets and settings

Our experiment aims to verify the recognition rate of NCBI methods for different types of clone blocks, effectively reduce the delay of BMEC systems and construct block chain networks. All algorithms are implemented using MATLAB R2018b on a 3.60 GHz computer with 64.0 memory.

Data sets: We download a data set with transaction information from the blockchain dataset website (http://xblock.pro/home-cn/), and then pre-process the data set to get our training set and test set.

Training set: In the first stage, we divide the data set into three subsets and carry out three different degrees of address markers. We mark heavy clone block dataset I with more than 66% error address information, moderately clone block dataset II with 33%–66% error information, lightly clone block dataset III with less than 33% error address information. Details of the processed datasets I, II, III are shown in Table 2. In the second stage, we use the processed datasets I, II, III for model training. We conduct 10 independent experiments on the training set, and then take the average recognition rate of 10 times as the result.

Test set: our train model is evaluated by the remaining blocks in the training set I, II, II. The recognition rate is verified when the number of neurons is 100–1000.

Table 2
Details of the datasets.

Datasets	Types	Training set	Verification set	Testing set	Total
Heavy clone block dataset	I	6865	2284	2285	11434
Moderately clone block dataset	II	8035	2678	2679	13392
Lightly clone block dataset	III	7192	2397	2398	11987

Settings: To test the performance of our proposed system, we create a three-layer neural network. The number of neurons in the input layer is set to 13, and the number of neurons in the hidden layer is set to 500.

We use the log-likelihood cost function for the training process, 3000 iterations, the learning rate $\eta = 0.1$, withdrawal from training with total error less than 0.008, and 10 times cross-validation is used to select the best optimization parameters.

We use a sharing model based on the Prim algorithm [7]. When the block is added to the blockchain network, the blockchain network is shared. We store the transaction information of the block itself in different small blocks, which are connected to each other The weight of the block is set by PoW, and then the minimum spanning tree is generated from the small block according to the Prim algorithm. In the MATLAB platform, we store the number of nodes of a blockchain network and the weights of interconnected edges in an array, and then divide the blockchain network.

When constructing the blockchain, too short a time slice of the time slice round-robin scheduling algorithm will lead to too many service switches and reduce the efficiency of the server, while too long a time slice may lead to poor response to short interactive requests. Therefore, setting the time slice to 100 ms is usually a reasonable compromise. We set 6 blocks to share the blockchain.

5.2. Experimental results and analysis

As shown in Fig. 6, we use our NCBI method to identify these three types of clone blocks. The NCBI method has the following characteristics for the recognition rate of three types of clone blocks: Type I has the highest recognition rate, Type II has the second highest recognition rate, and Type III has the lowest recognition rate. When the number of hidden layer neurons is 500, the recognition rate is the highest, which is due to the increase in the number of neuron nodes and overfitting which leads to a decrease in the recognition rate. However, as the number of neuron nodes increases, the recognition rate of clone blocks does not change significantly.

The result graph of neural network model training block nodes is shown in Fig. 7. Fig. 7(a) is the distribution of unknown blocks in blockchain network. Fig. 7(b) shows the block distribution map after the NCBI method is used. The blue node is the legal block, the yellow node is the heavy clone block, the green node is the moderately clone block, and the pink node is the lightly clone block, which clearly shows that our NCBI method can effectively identify some unknown blocks.

We set 10 blocks to sharing the blockchain, and the connection weight between each block is randomly assigned. After using the sharding method based on the Prim algorithm [7], the experimental results are shown in the Table 3. According to the experimental results, the comparison before and after the blockchain network division is shown in Fig. 8. In BMEC, as the number of block nodes in the edge server increases, after the blockchain is shared, the main chain and side chain transactions are distributed and executed in parallel, and each edge block node achieves load balancing during data processing.

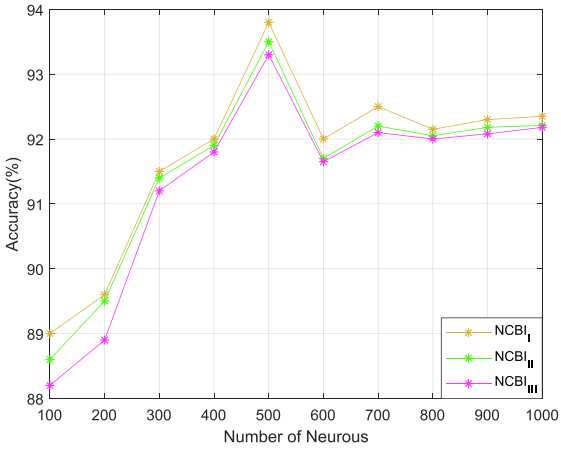
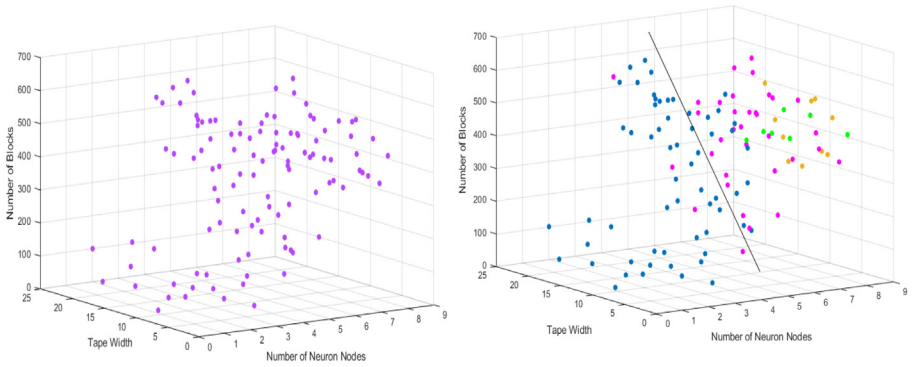


Fig. 6. Identification rate of different number of neurons.



(a) Distribution of original blocks.

(b) Neural network model training block distribution.

Fig. 7. Result graph of neural network model training block nodes.

Table 3
The Prim algorithm segmentation of blockchain network results.

Starting node	1	3	6	3	2
Next node	3	6	4	2	5
Weight	1	4	2	5	3

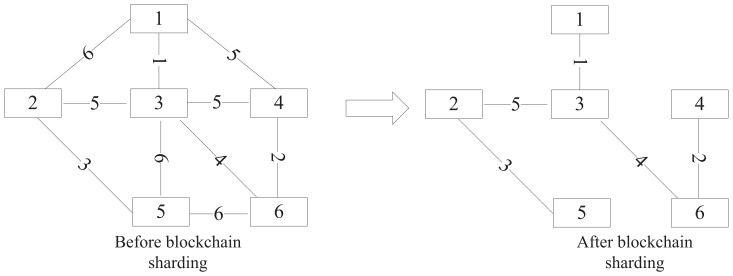


Fig. 8. Comparison before and after blockchain sharding based on the Prim algorithm.

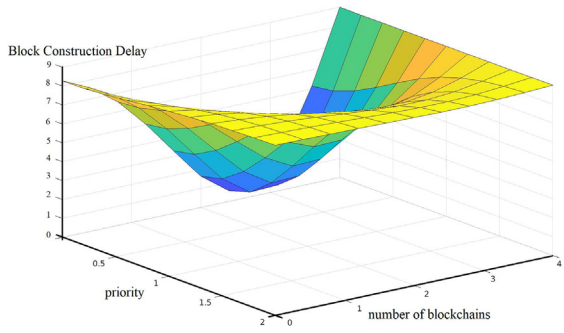


Fig. 9. Time slicing round-robin scheduling.

The larger the value of the time slice, the higher the priority of the block. Fig. 9 shows the relationships between the number of blockchains, the priority of the blocks, and the block construction delay when adding new blocks to the existing blockchain. In the experiments, we take 10 blocks to be added as an example, and execute the time slice round-robin scheduling algorithm [23] to determine the next block to be added. As shown in Fig. 10, as the number of blocks in the edge server increases, the total delay of blockchain construction in BMEC exhibits a non-linear increase. After sharing the blockchain,

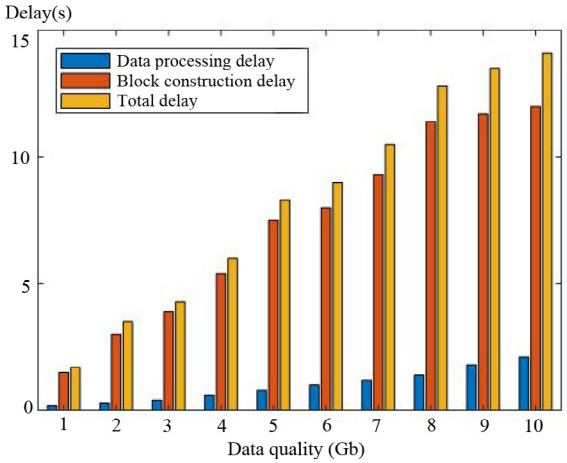


Fig. 10. Total delay of blockchain construction in BMEC.

the main chain and side chain transactions are distributed and executed in parallel, and each edge block server is performing data processing. This achieves load balancing and causes very little data processing delay. Thus, the total delay of the system mainly depends on the delay of the blockchain construction. To sum up, the experimental results in Figs. 9 and 10 show that our system can improve transaction throughput and have low system latency, which means that our system is scalable.

6. Conclusion

In this paper, we propose a blockchain-based mobile edge computing system (BMEC) that constructs each internet equipment into a block and connects each edge service node together to become a blockchain, the blockchain ensures the accuracy and consistency of data and rules in distributed edge devices. Then we propose a neural network based clone block identification (NCBI) method. After training the reliability metric of legitimate blocks and illegal clone blocks, we obtain a new neural network recognition model, and Furthermore, we abstract the edge server network topology into blockchain network topology, and then use the Prim algorithm [7] to divide it into the main chain (main storage block directory) and side chain (storage transaction details), which improves the data processing speed in the edge computing block system. In addition, the time slicing round-robin scheduling algorithm [23] is used to generate the blockchain, this method gives each block that wants to join the blockchain a priority, and determines the order of joining the blockchain according to the priority.

Noted that our system delay only considers data processing delay and block construction delay, and does not consider data transmission delay. In future work, we attend to add data transmission delay to the total delay and further optimize the proposed system. Further, we attend to develop novel algorithms to ensure the fairness and stability of the blockchain construction. Besides, we will apply our proposed system to solve other IoT problems.

7. Funding

This work is supported by the National Natural Science Foundation of China (61672321, 61771289, 61832012 and 11771251), Major Basic Research of Shandong Natural Science Foundation (ZR2019ZD10), Key Research and Development Plan of Shandong Province (2019GGX101050), Major agricultural application technology innovation project of Shandong Province (SD2019NJ007).

CRediT authorship contribution statement

Guangshun Li: Conceptualization, Methodology, Writing - original draft, Funding acquisition. **Xinrong Ren:** Methodology, Validation, Formal analysis, Writing - original draft. **Junhua Wu:** Supervision, Resources, Validation, Project administration. **Wanting Ji:** Investigation, Validation, Writing - original draft, Writing - review & editing. **Haili Yu:** Data curation, Writing - review & editing. **Jiabin Cao:** Visualization, Writing - review & editing. **Ruili Wang:** Investigation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Ali, S. Hassan, Y. Anupam, A dynamic metaheuristic optimization model inspired by biological nervous systems: neural network algorithm, Appl. Soft Comput. 71 (2018) 747–782, https://doi.org/10.1016/j.asoc.2018.07.039.

[2] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S.W. Cocco, J. Yellick, Hyperledger fabric: A distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, EuroSys '18, Association for Computing Machinery, New York, NY, USA, 2018, doi: 10.1145/3190508.3190538..

[3] S.D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain, in: Italian Conference on Cyber Security, doi: 10.5281/zenodo.1169273..

[4] Z. Cai, X. Zheng, J. Yu, A differential-private framework for urban traffic flows estimation via taxi companies, IEEE Trans Ind. Informat. 15 (2019) 6492–6499, https://doi.org/10.1109/TII.2019.2911697.

[5] Q. Chen, W. Wang, F. Wu, S. De, X. Huang, A survey on an emerging area: deep learning for smart city data, IEEE Trans. Emerg. Top. Comput. Intell. 3 (2019) 392–410, https://doi.org/10.1109/TETCI.2019.2907718.

[6] P. Danzi, A.E. Kalor, C. Stefanovic, P. Popovski, Delay and communication tradeoffs for blockchain systems with lightweight IoT clients, IEEE Internet Things J. 6 (2019) 2354–2365, https://doi.org/10.1109/JIOT.2019.2906615.

[7] R. Dondi, G. Mauri, I. Zoppis, Graph algorithms, in: S. Ranganathan, M. Gribskov, K. Nakai, C. Schönbach (Eds.), Encycl. Bioinform. Comput. Biol., Academic Press, Oxford, 2019, pp. 940–949, doi: 10.1016/B978-0-12-809633-8.20424-X..

[8] S. Fu, Q. Fan, Y. Tang, H. Zhang, X. Jian, X. Zeng, Cooperative computing in integrated blockchain-based Internet of Things, IEEE Internet Things J. 7 (2020) 1603–1612, https://doi.org/10.1109/JIOT.2019.2948144.

[9] K. Gai, Z. Fang, R. Wang, L. Zhu, P. Jiang, Kim-Kwang Raymond Chooc, Edge computing and lightning network empowered secure food supply management, IEEE Internet Things J. (2020) 1, https://doi.org/10.1109/JIOT.2020.3024694.

[10] K. Gai, Y. Wu, L. Zhu, Z. Zhang, M. Qiu, Differential privacy-based blockchain for industrial Internet-of-Things, IEEE Trans Ind. Informat. 16 (2020) 4156–4165, https://doi.org/10.1109/TII.2019.2948094.

[11] J. Li, J. Wu, L. Chen, Block-secure: Blockchain based scheme for secure p2p cloud storage, Inf. Sci. 465 (2018) 219–231, https://doi.org/10.1016/j.ins.2018.06.071.

[12] C. Liu, Y. Xiao, V. Javangula, Q. Hu, S. Wang, X. Cheng, Normachain: A blockchain-based normalized autonomous transaction settlement system for iot-based e-commerce, IEEE Internet Things J. 6 (2019) 4680–4693, https://doi.org/10.1109/JIOT.2018.2877634.

[13] Z. Ma, X. Wang, J. Deepak Kumar, K. Haneef, H. Gao, Z. Wang, A blockchain-based trusted data management scheme in edge computing, IEEE Trans Ind. Informat. 16 (2020), https://doi.org/10.1109/TII.2019.2933482.

[14] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, Y. Zhao, EdgeChain: An Edge-IoT framework and prototype based on blockchain and smart contracts, IEEE Internet Things J. 6 (2019) 4719–4732, https://doi.org/10.1109/JIOT.2018.2878154.

[15] L. Qi, R. Wang, C. Hu, S. Li, Q. He, X. Xu, Time-aware distributed service recommendation with privacy-preservation, Inf. Sci. 480 (2019) 354–364, https://doi.org/10.1016/j.ins.2018.11.030.

[16] S. Sanhua, Reliability metric-based detection algorithm of compromised sensor nodes in wireless sensor networks, China Meas. Test. 44 (2018) 148–152. 10.11857/j.issn.1674-5124.2018.07.028..

[17] P.K. Sharma, S. Rathore, Y. Jeong, J.H. Park, SoftEdgeNet: SDN based energy-efficient distributed network architecture for edge computing, IEEE Commun. Mag. 56 (2018) 104–111, https://doi.org/10.1109/MCOM.2018.1700822.

[18] L. Tian, J. Li, W. Li, B. Ramesh, Z. Cai, Optimal contract-based mechanisms for online data trading markets, IEEE Internet Things J. 6 (2019) 7800–7810, https://doi.org/10.1109/JIOT.2019.2902528.

[19] Z. Tian, M. Li, M. Qiu, Y. Sun, S. Su, Block-def: A secure digital evidence framework using blockchain, Inf. Sci. 491 (2019) 151–165, https://doi.org/10.1016/j.ins.2019.04.011.

[20] D. Wang, C. Cui, Stochastic configuration networks ensemble with heterogeneous features for large-scale data analytics, Inf. Sci. 417 (2017) 55–71, https://doi.org/10.1016/j.ins.2017.07.003.

[21] D. Wang, M. Li, Stochastic configuration networks: fundamentals and algorithms, IEEE Trans. Syst., Man, Cybern. 47 (2017) 3466–3479, https://doi.org/10.1109/TCYB.2017.2734043.

[22] L. Wang, X. Zhang, R. Wang, C. Yan, L. Qi, Diversified service recommendation with high accuracy and efficiency, Knowl-based Syst. 204 (2020), https://doi.org/10.1016/j.knosys.2020.106196 106196.

[23] J. Xiao, X. Zhang, An improved time slice round-robin scheduling algorithm, Comput. Appl. 25 (2005) 447–448, JournalArticle/5af17869c095d718d8e6c7b0.

[24] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, Z. Han, When mobile blockchain meets edge computing, IEEE Commun. Mag. 56 (2018) 33–39, https://doi.org/10.1109/MCOM.2018.1701095.

[25] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, M.Z.A. Bhuiyan, Joint optimization of offloading utility and privacy for edge computing enabled IoT, IEEE Internet Things J. 7 (2020) 2622–2629, https://doi.org/10.1109/JIOT.2019.2944007.

[26] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, M.Z.A. Bhuiyan, Joint optimization of offloading utility and privacy for edge computing enabled iot, IEEE Internet Things J. 7 (2020) 2622–2629, https://doi.org/10.1109/JIOT.2019.2944007.

[27] Z. Yang, K. Yang, L. Lei, K. Zheng, V.C.M. Leung, Blockchain-based decentralized trust management in vehicular networks, IEEE Internet Things J. 6 (2019) 1495–1505, https://doi.org/10.1016/j.ins.2018.06.071.

[28] C. Yao, X. Wang, Z. Zheng, G. Sun, L. Song, EdgeFlow: Open-source multi-layer data flow processing in edge computing for 5G and beyond, IEEE Netw. 33 (2019) 166–173, https://doi.org/10.1109/MNET.2018.1800001.

[29] J. Zhang, J. Hu, J. Liu, Neural network with multiple connection weights, Pattern Recognit. 107 (2020), https://doi.org/10.1016/j.patcog.2020.107481 107481.

[30] S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, Efficient knn classification with different numbers of nearest neighbors, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 1774–1785, https://doi.org/10.1109/TNNLS.2017.2673241.

[31] Y. Zhang, R.H. Deng, X. Liu, D. Zheng, Blockchain based efficient and robust fair payment for outsourcing services in cloud computing, Inf. Sci. 462 (2018) 262–277, https://doi.org/10.1016/j.ins.2018.06.018.

[32] X. Zheng, Z. Cai, J. Li, H. Gao, A study on application-aware scheduling in wireless networks, IEEE Trans. Mobile Comput. 16 (2017) 1787–1801, https://doi.org/10.1109/TMC.2016.2613529.

[33] W. Zhong, X. Yin, X. Zhang, S. Li, W. Dou, R. Wang, L. Qi, Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment, Comput. Commun. 157 (2020) 116–123, https://doi.org/10.1016/j.comcom.2020.04.018.

[34] T. Zhu, T. Shi, J. Li, Z. Cai, X. Zhou, Task scheduling in deadline-aware mobile edge computing systems, IEEE Internet Things J. 6 (2019) 4854–4866, https://doi.org/10.1109/JIOT.2018.2874954.