



# An efficient privacy-preserving blockchain storage method for internet of things environment

Dayu Jia<sup>1</sup> · Guanghong Yang<sup>1,2</sup> · Min Huang<sup>1,2</sup> · Junchang Xin<sup>3,4,5</sup> · Guoren Wang<sup>6</sup> · George Y. Yuan<sup>6,7</sup>

Received: 28 January 2023 / Revised: 25 March 2023 / Accepted: 28 March 2023 /  
Published online: 8 May 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Blockchain is a key technology to realize decentralized trust management. In recent studies, sharding-based blockchain models are proposed and applied to the resource-constrained Internet of Things (IoT) scenario, and machine learning-based models are presented to improve the query efficiency of the sharding-based blockchains by classifying hot data and storing them locally. However, in some scenarios, these presented blockchain models cannot be deployed because the block features used as input in the learning method are privacy. In this paper, we propose an efficient privacy-preserving blockchain storage method for the IoT environment. The new method classifies hot blocks based on the federated extreme learning machine method and saves the hot blocks through one of the sharded blockchain models called ElasticChain. The features of hot blocks will not be read by other nodes in this method, and user privacy is effectively protected. Meanwhile, hot blocks are saved locally, and data query speed is improved. Furthermore, in order to comprehensively evaluate a hot block, five features of hot blocks are defined, including objective feature, historical popularity, potential popularity, storage requirements and training value. Finally, the experimental results on synthetic data demonstrate the accuracy and efficiency of the proposed blockchain storage model.

**Keywords** Blockchain · Federated learning · Extreme learning machine · Privacy protection · Edge nodes

## 1 Introduction

Internet of things (IoT) [1] and Blockchain [2] can be merged for better results which leads us to one level up in many fields, such as agriculture [3], supply chain [4] and healthcare [5]. In these different fields, by using IoT technology, the process of collecting data is accurate

---

✉ Dayu Jia  
jiadayu@mail.neu.edu.cn

Extended author information available on the last page of the article

and complete, and blockchain technology ensures that the storage of the collected data is decentralized and tamper-resistant.

Therefore, in recent years, many studies [6–10] have been conducted to improve the performance of blockchain systems in IoT scenarios from various aspects, such as improving storage scalability and query efficiency. One of the studies proposed a storage scalable blockchain model, called ElasticChain [6], in which full nodes store partial data fragments of a complete blockchain based on a duplicate ratio regulation algorithm.

Based on the research [6], in order to ensure the security of fragmented data, [7] presented a comprehensive evaluation method of node reliability based on Extreme learning machine (ELM) to distribute fragmented data to reliable nodes. Meanwhile, for the sake of efficient queries, [8] proposed an optimal data distribution scheme based on ELM to distinguish the hot blocks for storage. ELM [11] is a machine learning method with high training and classification performance. Some private data of user nodes, such as transaction records, read records and node performance information, are read by the ELM classifiers as features to distinguish reliable nodes.

Nodes where ELM classifiers are deployed are generally not user nodes, which leads to the security risk of user privacy data being stolen. For example, a node with permission to read user private data can launch a reconstruction attack [12] to reuse user data or launch an attribute-inference attack [13] to know the real identity of the data owner. These security risks will seriously harm the interests of users, especially during the current boom of web 3.0 [14]. Currently, many countries have attached great importance to privacy protection. For example, the European Union implemented the General Data Protection Regulation (GDPR) [15] in 2018, and the California consumer privacy Act (CCPA) [16] came into effect in 2020. Therefore, it is of great significance to realize the privacy protection of blockchain user data in the IoT environment under the premise of data security and efficient query.

Scenario. In some practical scenarios, such as the Internet of Medical Things (IoMT) and wise medical, the patient's treatment information or vital-sign data are highly sensitive, and these data cannot be shared with other people or medical institutions. In these scenarios, if the privacy of user data cannot be protected, efficient blockchain systems such as [7, 8] cannot be applied.

Based on the current methods, the above privacy protection problems in the blockchain systems [7, 8] cannot be well solved. In order to achieve privacy protection, the current methods that can be used are to delete the sensitive data for evaluating the reliability of nodes or discard the sensitive data in the features of evaluating hot nodes. These methods will significantly degrade the performance of blockchain models. Therefore, this paper proposes a blockchain storage method based on federated learning to achieve fast query under the premise of protecting data privacy in the Internet of Things environment.

This paper first introduces the architecture of this new model and designs a federated extreme learning machine-based hot block storage strategy to achieve fast query and data privacy protection. The proposed architecture applies to the resource-constrained environment, where the nodes store only part of blocks. In the model, blocks are divided into hot blocks and non-hot blocks for a node according to the block features, and the hot blocks will be saved in this node locally. When the query is initiated, the query will probably be performed locally without querying other nodes, thus improving the query efficiency. Meanwhile, the federated ELM-based hot block storage strategy ensures that private feature data will not be read by other nodes during training. Then, in order to comprehensively evaluate a hot block, this paper defines five features of hot blocks: objective feature of blocks, historical popularity of blocks, potential popularity of blocks, storage requirements and training value.

Furthermore, an incentive mechanism is designed in this paper to motivate nodes to actively participate in the proposed model and provide valuable data.

In summary, the contributions of this paper include the following:

- We propose a hot block storage model for sharded blockchain systems to achieve fast query and data privacy protection.
- We design a comprehensive and objective evaluation method to classify the hot blocks for blockchain nodes.
- We conduct a set of experiments to demonstrate the accuracy of the hot block storage model and the query efficiency of the blockchain system using the storage model based on synthetic data.

The rest of this paper is organized as follows. Section 2 reviews related works before Section 3 provides relevant background. Section 4 describes our approach in detail, and Section 5 reports experimental evaluation. Finally, we conclude in Section 6.

## 2 Related work

### 2.1 Blockchain-enabled internet of things

An architecture of blockchain-empowered edge intelligence for IoMT is presented [17] to offer solutions for COVID-19, such as monitoring and tracing the pandemic origin, traceable supply chain of vaccines and remote healthcare services. A new privacy-preserved threat intelligence framework (P2TIF) is introduced [18] to protect confidential information and to identify cyber-threats in the Industrial Internet of Things (IIoT) environments. A scalable blockchain module is adopted to ensure secure communication of IIoT data and prevents data poisoning attacks. A Privacy-Preserving and Secure Framework (PPSF) is proposed for IoT-driven smart cities [19]. A blockchain module is used in a two-level privacy scheme to securely transmit the IoT data and the Principal Component Analysis (PCA) technique is applied to transform raw IoT information into a new shape. An optimal smart contract [20] is designed for autonomous greenhouse environment based on IoT blockchain network in agriculture. There are three steps to provide optimal greenhouse environment: prediction, optimization, and finally controlling. Thus, it can be seen that the research of blockchain technology has been widely used in many fields such as IoMT, IIoT, smart cities and agricultural IoT, etc.

### 2.2 Privacy protection in IoT

A secure fingerprint authentication system [21] is proposed to protect user privacy for authentication on IoT devices. A weight-based fusion mechanism and a linear convolution-based transformation are used to increase security and protection. A Federated Edge Aggregator (FEA) framework with differential privacy is introduced [22] for safeguarding healthcare industry using IoT systems. An iteration-based converged Convolutional Neural Network model is developed to perform edge intelligence to balance FL's privacy preservation. A modified ciphertext policy attribute-based encryption (CP-ABE) scheme with a constant length of ciphertext is presented [23] to reduce computation overhead in the encryption and decryption phases. In this scheme, chameleon hash and bloom filters are adopted to prevent authorized users from leaking data and protect the privacy of data owners. A blockchain-based privacy protection scheme is designed [24] for IoT-assisted educational big data management. The

full system is implemented, including the components of collecting and storing the educational data and maintaining a role-based access control on educational data.

Thus, it can be seen that technologies such as fingerprint authentication, federated learning, encryption algorithms and blockchain are applied to achieve privacy protection in IoT. However, the problem of data privacy protection in IoT scalable blockchain still exists, and an efficient solution is proposed based on existing methods.

### 2.3 The combination of federated learning and blockchain

A reliable federated learning system, called Refiner [25], is proposed for tackling the challenges introduced by massive engagements of self-interested and malicious participants. In Refiner, an incentive mechanism is built to engage self-interested participants based on Ethereum. A blockchain-based federated learning application trading platform called FLeX is introduced [26]. In FLeX, FL algorithms and trading mechanisms are deployed in smart contracts of the blockchain to ensure data security. A blockchain-based asynchronous federated learning (BAFL) framework is designed [27] to ensure the security and efficiency required by FL. The asynchronous learning speeds up global aggregation, and blockchain ensures that the model data cannot be tampered with. A secure sharing method of Marine Internet of Things (MIoT) data is presented [28] under an edge computing framework. Federated learning is used to ensure the privacy of nodes and the blockchain serves as a decentralized way, which stores federated learning workers to achieve non-tampering and security.

Thus, it can be seen that many studies combine blockchain and federated learning to ensure data security and data privacy. However, most of the studies are based on federated learning models and use blockchain technology to ensure the data security of the model or incentivize node participation. This paper is based on the blockchain system, using federated learning technology to achieve data privacy protection in the blockchain system.

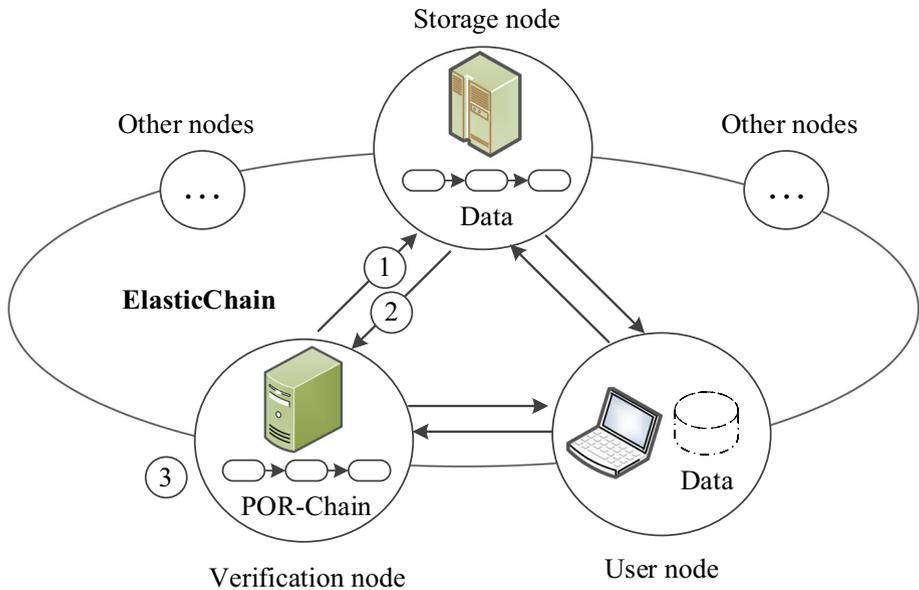
## 3 Preliminaries

### 3.1 IoT-enabled storage scalable models for blockchain

ElasticChain [6] is a storage scalable blockchain model, which can be deployed to edge nodes in IoT. The main structure of ElasticChain is shown in Figure 1. There are three roles in ElasticChain: user node, storage node and verification node. User nodes are the owner of the data. The transactions are produced between user nodes. Storage nodes are used to save the fragmented blockchain data, and the verification nodes are responsible for finding reliable storage nodes according to the node reliability verification method.

The process of node reliability verification method is shown as Step1-3 in Figure 1. Firstly, verification nodes detect the reliability of data in storage nodes at the same period of time, and store the reliability value of storage nodes in the POR (Proofs of Reliability) chain. Then, the reliable storage nodes are singled out to store blockchain data.

However, one of the shortcomings in ElasticChain is that the features of evaluating node reliability are incomplete. Data may be stored in unreliable nodes resulting in reduced security. To solve this shortcoming, [7] defines five new evaluation indicators to distinguish reliable nodes. The indicators are the security, trustworthiness, activeness, stability and communication costs of storage nodes. Meanwhile, the ELM classifier is used to improve classification efficiency.



**Figure 1** The structure of ElasticChain

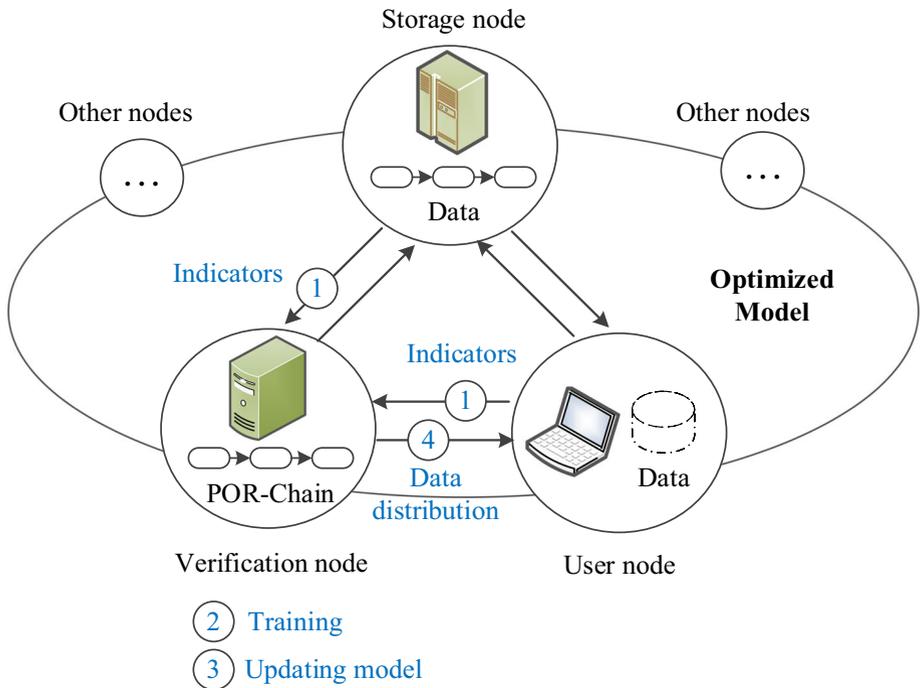
The data processing process of [7] is shown in Figure 2. First, verification nodes read the indicators from user node and storage node. Then, the classifier is trained based on the indicators. After training, the new model is updated and used to select reliable storage nodes.

The other shortcoming of ElasticChain is that reliable nodes store non-hot data, which reduces query efficiency. To solve this shortcoming, [8] proposes a new ELM-based method to distinguish hot blocks according to five aspects: the objective evaluation of the block, the objective evaluation of the block related to the node, the historical popularity of the block being used by the node, the hidden popularity of the block and the storage requirements of the block. The data processing process of [8] is similar to that of Figure 2. The data read by the verification nodes is the evaluation indicator of the hot block.

### 3.2 The theory of federated extreme learning machine system (Fed-ELMS)

ELM [29] is originally developed for single hidden-layer feedforward neural networks (SLFNs) and can achieve better generalization performance than other conventional learning algorithms at an extremely fast learning speed [30]. Therefore, ELM is selected as the classification method in this paper. Moreover, to meet the demand for privacy-preserving federated learning scenarios, a federated extreme learning machine system (Fed-ELMS) is proposed in [31]. Fed-ELMS can achieve comparable performance with only 3.3% accuracy loss compared to a centralized ELM.

Fed-ELMS can be divided into two parts, centralized ELM server and local ELM on edge devices [31]. In centralized ELM server,  $\text{Max}(\lfloor C \times K \rfloor, 1)$  clients are selected first.  $K$  is the number of clients,  $C$  is the fraction of chosen client. The selected edge device will receive the output weight  $\beta$  and meta weight  $P$  of the global ELM model. Each edge device will



**Figure 2** The structure of optimized model based on ElasticChainlabel

send the updated output weights, meta weights and the number of data used in each selected edge device to the global server.  $\beta$  and  $P$  are updated as the following formula.

$$\beta \leftarrow \sum_{k=1}^{Max(\lfloor C \times K \rfloor, 1)} \frac{N_k}{\sum_{k=1}^{Max(\lfloor C \times K \rfloor, 1)} N_k} \beta^k \tag{1}$$

$$P \leftarrow \sum_{k=1}^{Max(\lfloor C \times K \rfloor, 1)} \frac{N_k}{\sum_{k=1}^{Max(\lfloor C \times K \rfloor, 1)} N_k} P^k \tag{2}$$

Here,  $\leftarrow$  denotes a sequential, or non-concurrent assignment.  $\beta^k$  and  $P^k$  are output weights and meta weights from the  $k$ th edge device.

In the part of local edge devices, devices receive parameters  $\beta$  and  $P$  from the central server and then update their local OS-ELM model. The parameters  $\beta^k$  and  $P^k$  in edge devices are updated as the following formula.

$$\beta^k = \beta + P^k H^T (T - H \beta) \tag{3}$$

$$P^k = P - P H^T (I + H P H^T)^{-1} H P \tag{4}$$

Here,  $H$  is the hidden layer output matrix of the existing data ( $X^k = \{(x_i, t_i)\}_{i=1}^{N^k}$ ).  $T$  is the matrix form of all the target vectors  $t_i$ . After updating the local ELM, the edge devices will upload the updated parameters  $\beta^k$ ,  $P^k$ , and  $N^k$  to the centralized server to be used in updating the global model.

## 4 Federated learning-based blockchain storage model

In this section, we first describe the architecture of the federated learning-based (FL-based) blockchain storage model. Then we introduce the features used to classify reliable nodes and hot blocks. After that, the algorithm of data distribution process and node incentive mechanism are proposed.

### 4.1 Architecture

In order to realize that the blockchain system can not only accurately classify hot blocks for storage through machine learning methods, but also ensure that the private data of each storage node will not be read by other nodes during training, federated learning method is adopted in this paper. Figure 3 shows the architecture of the FL-based blockchain storage model.

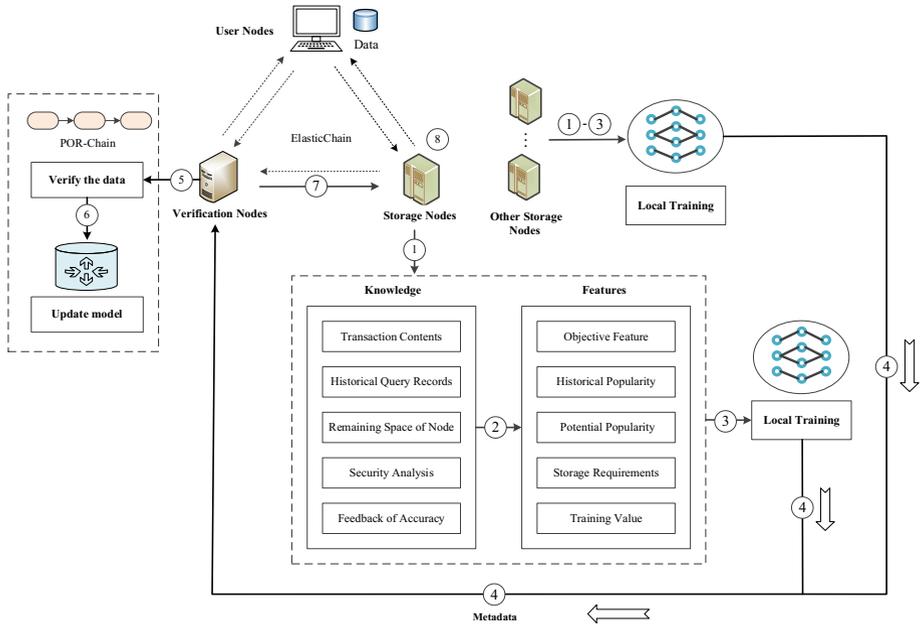
Nodes in the new model still have 3 roles: user node, storage node and verification node. The user nodes are the producer of the blockchain data. The generated data fragments are stored in the storage nodes. The verification nodes detect the reliability of the storage nodes in real time and return the reliability value to the user nodes, so as to realize the fragmented data of the blockchain is stored in the reliable storage node and ensure data security.

Moreover, in the FL-based blockchain storage model, each storage node locally trains the hot block classification model and sends the trained gradients to the validation nodes. After receiving the gradients from the storage nodes, the verification nodes will first verify the authenticity of the gradients. Then, based on the new gradients, an accurate model is aggregated and sent to each storage node. The storage nodes will classify the non-hot blocks according to the newly received classification model and delete the non-hot blocks when their storage space is insufficient.

Specifically, the storage node reads five pieces of knowledge (transaction contents, historical query records, remaining space of node, security analysis and feedback of accuracy) of the local block at a fixed interval. Then, five features of hot blocks are calculated based on the five pieces of knowledge, which are the objective feature of blocks, historical popularity of blocks, potential popularity of blocks, storage requirements and training value, respectively. At the same time, storage nodes record the usage frequency of each block in a fixed interval. Then, local ELM-based training is performed locally on each storage node to obtain model gradients in the new time interval. Finally, the new gradient is sent to the validation nodes.

The verification nodes first verify the authenticity of the uploaded gradient after receiving the gradient information sent by the storage nodes. The gradient verification methods based on federated learning can be used, such as FLDetector [32]. If the verification result is true, the correct gradient will be recorded in the POR chain, and a new hot block classification model will be aggregated according to the centralized ELM method and sent to each storage node. The new model will be sent to each storage node. If it is verified that the received gradient data is malicious, such as uploading unreal gradient data to reduce the classification accuracy of the aggregation model, the malicious behavior of the storage nodes will be recorded in the POR chain.

The purpose of writing gradient updates to the POR chain is to ensure that each record is immutable and traceable. This effectively prevents malicious node behavior. Meanwhile, the incentive mechanism for storage nodes to upload valid gradient data is designed in Section 4.4. In addition, since the POR chain already exists in the Elasticchain model to



**Figure 3** The architecture of FL-based blockchain storage model

record the reliability value of the storage nodes, writing the gradient update data to the POR chain will not cause a lot of additional storage overhead.

In the FL-based blockchain storage model, the storage nodes effectively protect the privacy security of their own data because they do not send their private data directly to the verification nodes. At the same time, the data of each storage node is fully utilized in training, so an accurate classification model of hot blocks is obtained. As a result, the new model realizes that the hot blocks are classified and stored locally. When querying data, more local queries are required instead of initiating query requests to other nodes, which improves query efficiency.

### 4.2 Feature selection

Five features are chosen to classify the hot blocks. Each selected feature is calculated from other two to four different features. The dimensionality of the classifier is reduced by summarizing several related features into one feature, which improves the classification speed. Then, the five features of a hot block are defined below.

**Definition 1** (Objective feature of blocks (*OF*)) The objective feature of a block is fixed after the block is generated. The value of *OF* is evaluated in two aspects: the recognized objective value of the block and the value of the block to the node where it is stored. The recognized objective value can be evaluated by the number and value of transactions in the block. When the sum of transaction values contained in the block is large, or there are a great number of transactions in a block, the objective eigenvalue of the block is large. In addition, in some cases, although a block has a large recognized objective value, it has little relevance to a storage node. This storage node may rarely access or reads the contents of this block. Therefore, it is also necessary to consider the value of a block for the node that stores it.

When a block contains  $T$  transactions and the value of transactions is  $\{V_1, V_2, \dots, V_T\}$ ,  $OF$  can be expressed as follows:

$$OF = \omega T \times \sum_{i=1}^T (V_i) + (1 - \omega) \left( \frac{\sum_{i=1}^T T b_i}{T} \times \sum_{i=1}^T \frac{V b_i}{V_i} \right) \tag{5}$$

Here,  $\omega$  ( $0 \leq \omega \leq 1$ ) is the weight of the recognized objective value of the block. The larger the value of  $\omega$ , the greater the proportion of the recognized objective value of the block in  $OF$ . Conversely, the greater the weight of the value of the block to the storage node. Besides, if the initiator or receiver of the  $i$ th transaction in the block is the storage node of this block,  $T b_i = 1$  and  $V b_i = V_i$ , otherwise  $T b_i = 0$  and  $V b_i = 0$ .

**Definition 2** (Historical popularity of blocks ( $HP$ )) The historical popularity of a block is also an important indicator for evaluating a hot block. The value of  $HP$  is evaluated by the total number of times the block is accessed by the storage node and the time of access. The more times a block is accessed, the greater is the  $HP$  value for that block. Meanwhile, if the time when a block is accessed by the storage node is close to the current time, it means that the block is the hot data for this node, and its  $HP$  value is larger. In order to reduce the extra cost caused by counting the number of accesses, the fixed time interval  $S$  is set. The total number of times the node reads the block is recorded at a fixed interval time  $S$ . Assume that  $J$  time intervals have been generated ( $S_1$  to  $S_J$ ). In the  $S_j$  interval ( $1 \leq j \leq J$ ), the total number of times the storage node reads the block is  $N_j$ , and the number of times that other nodes query the block data from the storage node is  $No_j$ .  $HP$  can be expressed as follows:

$$HP = \sum_{j=1}^J (N_j + \alpha No_j) \times \frac{1}{J - j + 1} \tag{6}$$

Here,  $\alpha$  ( $0 \leq \alpha \leq 1$ ) is the weight of the total access times of other nodes. The larger the value of  $\alpha$ , the more important the node’s contribution to the whole system.  $1/(J - j + 1)$  represents the distance between the time interval and the current time.

**Definition 3** (Potential popularity of blocks ( $PP$ )) It is not accurate that *Block-A* is considered a non-hot block of *Node-1* when *Block-A* does not save the transactions participated by *Node-1* (based on Definition 1) in some cases. The reason is that the data stored in *Block-A* may be highly correlated with node *Node-1*. For example, the participants of many transactions ( $Transaction_1 - Transaction_r$ ) are *Node-1* and *Node-2*. If the participants of all transactions in *Block-A* are not *Node-1* and  $Transaction_1 - Transaction_r$  are not saved in *Block-A*, but a large number of transactions involving *Node2* are stored in *Block-A*, *Node-1* will most likely read *Block-A* to assist other nodes in verifying  $Transaction_1 - Transaction_r$ . Therefore, *Block-A* is considered to have high potential popularity with *Node-1*. When *Block-A* contains more transactions related to *Node-2* and the transaction value is larger, the potential popularity value of *Block-A* for *Node-1* is higher.  $PP$  can be expressed as follows:

$$PP = \sum_{i=1}^Y V_y \times (\rho + \mu) \times \frac{1}{T_y} \tag{7}$$

Here, the target node generates  $Y$  transactions. In the  $y$ th transaction ( $1 \leq y \leq Y$ ), the node that is trading with the target node is  $Node_y$ , and the interval between the transaction

time and the current time is  $T_y$  ( $T_y$  is an upward integer in hours, with a minimum value of 1). There are  $T$  transactions in the target block, and the value of each transaction is  $V_y$ . If the initiator of the  $t$  transaction is one node of  $Node_1 - Node_Y$ ,  $\rho = 1$ , otherwise  $\rho = 0$ . If the receiver of the  $t$  transaction is one node of  $Node_1 - Node_Y$ ,  $\mu = 1$ , otherwise  $\mu = 0$ .

**Definition 4** (Storage requirements ( $SR$ )) The storage requirement of a block is an important indicator for nodes to determine whether to save the block. The storage requirements of a block can be evaluated from three aspects. The first is the security of the block. In blockchain, the hash value of a new block is calculated based on the hash value of the previous block. As a result, it is less difficult for an attacker to tamper with the hash of a new block than it is for an old block. Therefore, according to the security of the block, the new block needs to be saved with a larger number of copies, while the number of old block copies can be appropriately reduced. Second, we need to consider the difference between the number of copies that have been saved in the system and the number of copies that need to be saved. If the number of copies of a block that has been saved is far less than the required number, the value of the block storage requirement is large. Third, when the remaining space of the node is enough, the block can be saved locally.

$$SR = \lambda \frac{B_l}{B_n} \times \delta \frac{S_r - S_b}{S_r} \times \begin{cases} (D_n - D_e) & D_n > D_e \\ 1 & D_n \leq D_e \end{cases} \tag{8}$$

Here,  $B_l$  is the number of the block and the first generated block is number 1.  $B_n$  is the number of the latest block.  $S_b$  is the storage space required by the block, and  $S_r$  is the remaining storage space in the node. In most cases,  $S_r \geq S_b$  and the value of  $SR$  is calculated according to the above formula. When  $S_r < S_b$ , the node does not have enough space to save the block.  $D_e$  is the number of copies of the block that have been saved in the system, and  $D_n$  is the number of copies that the block needs to store.  $\lambda$  is the weight of block security and  $\delta$  is the weight of the remaining storage space.

**Definition 5** (Training value ( $TV$ )) After the block data is locally trained and the model is updated, other nodes will use the new model to distinguish hot blocks for storage. The training value can be understood as the benefit of the new model to user nodes. The greater the benefit, the higher the training value of the block data. In order to encourage nodes to contribute their data in the federated learning model, the model will evaluate the benefits brought by nodes to the model and give corresponding rewards. When one node provides data with a high training value, each node will receive corresponding rewards when other nodes gain benefits. Therefore, nodes prefer to save blocks with high training values locally. The blocks with high training values are set with higher heat values. In the federated learning model, many computational contributions and incentive methods have been proposed, such as sharing game [33] and reverse auction [34]. The effectiveness of these methods has been verified and can be used in the training value evaluation of this model.

### 4.3 FL-based blockchain storage model

The hot block storage strategy in the FL-based blockchain storage model is shown in Algorithm 1. First, each storage node in the system records the knowledge of local blocks after a fixed time interval, including  $T, V_i, T b_i, V b_i, J, N_j, N o_j, Y, V_y, T_y, B_l, B_n, S_r, S_b, D_n$  and  $D_e$ . Then, each storage node sets the parameters ( $\omega, \alpha, \rho, \mu, \lambda$  and  $\delta$ ) of hot blocks according to the implementation requirements. Next, according to the knowledge and parameters of the blocks, the storage nodes calculate the five features of each block(objective feature of

blocks ( $OF$ ), historical popularity of blocks ( $HP$ ), potential popularity of blocks ( $PP$ ), storage requirements ( $SR$ ) and training value ( $TV$ ). After that, each storage node reads the usage frequency of the blocks in the fixed time interval and the local ELM is trained on each storage node based on the block features and the usage frequency. The model gradient in the new time interval is obtained and sent to the verification node. After the verification node receives the new gradient, the authenticity of the new gradient will be verified. If the gradient is fabricated, the malicious behavior is recorded in the POR chain, and then the information of the node is masked. If the gradient data is real, the gradient information is written into the POR chain. Then a new hot block classification model is aggregated according to the centralized ELM method and sent to each storage node. The storage nodes classify the hot blocks according to the new model and store them locally.

---

**Algorithm 1** The hot block storage strategy.

---

**Input:** records of the blocks usage and feature evaluation parameters;

**Output:** storage scheme of the block;

- 1 storage nodes ( $N_S$ ) record the knowledge of blocks after a fixed time interval;
  - 2  $N_S$  set the parameters ( $\omega, \alpha, \rho, \mu, \lambda$  and  $\delta$ );
  - 3  $N_S$  calculate the features of blocks ( $OF, HP, PP, SR$  and  $TV$ );
  - 4  $N_S$  read the usage frequency of the blocks in the fixed time interval;
  - 5 each  $N_S$  train the local ELM model based on the block features and the usage frequency, and the new gradient is obtained;
  - 6 each  $N_S$  send the new gradient to verification nodes ( $N_V$ );
  - 7  $N_V$  receive the new gradients ( $G$ ) and verify the authenticity of the gradients;
  - 8 **If** ( $G$  is false)
  - 9      $N_V$  record the malicious behavior in the POR chain;
  - 10    the  $G$  from the malicious nodes will be masked in the future;
  - 11 **Else;**
  - 12     $G$  is written into the POR chain;
  - 13 aggregate a new hot block classification model according to the centralized ELM method;
  - 14  $N_V$  send the new model to each  $N_S$ ;
  - 15  $N_S$  classify the hot blocks based on the latest model;
  - 16  $N_S$  store hot blocks locally;
  - 17 **End If;**
- 

#### 4.4 Incentive mechanism

In the FL-based blockchain storage model, a large number of storage nodes are expected to maintain the blockchain ledger. However, the storage node occupies its own storage space to save the block data, and consumes the node performance for local model training. Therefore, an effective incentive mechanism is necessary to let more storage nodes join the system.

In the model, storage nodes evaluate the performance of the hot block classification model sent by the verification node every time interval. Compare the results of the classification model with the actual utilization of the block. Then the evaluation results are returned to the verification node. If the average performance of classification models from different storage nodes during this interval is improved compared with the past, the verification node will find the gradient data used to train this model. Then, the storage nodes that provide these gradient data will be rewarded.

In terms of reward methods, one kind of reward can be virtual currency, such as Bitcoin and Ether. These virtual currencies come from the users of the system. Reasonable charges will

be accepted by users because the system provides users with a privacy-preserving blockchain storage platform. After charging a certain fee according to the user's data storage capacity, the system will reward the nodes that provide the gradient based on the contribution value of the gradient data. Another way of reward is that the system gives some storage space to nodes that provide valid gradient data so that these nodes can store more data in the system.

Through the incentive mechanism, storage nodes will actively participate in the system and provide their own storage space and node performance for training. Meanwhile, the storage nodes are encouraged to provide effective local training data, and the useless training data is reduced.

## 5 Evaluation

First, Section 5.1 provides the setup of evaluations. Then, Section 5.2 discusses the evaluation of the FL-based classifier. Finally, Section 5.3 shows the evaluation of the query performance in ElasticChain.

### 5.1 Experiment settings

All experiments are set up on a desktop computer with Intel Core i5 CPU and 16GB RAM, running on Windows 7. The nodes in the FL-based blockchain storage model are all set up on a VMware workstation with 1GB hard disk space and 300MB RAM, running on ubuntu 16.04.

There are 30 nodes in the experiment and each node has 50 different sets of parameters. Therefore, there are 1500 sets of feature data, and each set of data is divided into hot block or non-hot block.

The synthetic data are used as the experimental data set. The controlled variable method is used to set different parameters for blocks according to the real application situation. For example, we take the data of Bitcoin system as a reference. The number of transactions in each block and the transaction value in the block are read [35], and the number of active addresses when the block is generated is used as an indicator to evaluate the block heat. The real data is not used as the experimental data set because the existing real data set cannot meet the requirements of this experiment. Currently, blockchain-based wise medical systems are applied in some hospitals or medical institutions. However, in order to protect the privacy of patients, the data has not been published. Meanwhile, we can download some features of blocks (objective feature of blocks ( $OF$ )) in some public blockchain platforms, such as Ethereum and Bitcoin, but other block features ( $HP$ ,  $PP$ ,  $SR$  and  $TV$ ) can not be downloaded.

The four feature values ( $OF$ ,  $HP$ ,  $PP$  and  $SR$ ) are calculated by the formula (5), (6), (7) and (8), respectively. In formula (5), the value of  $OF$  is greatly affected by  $V_i$ , so  $V_i$  is set to a value with a large difference ( $V_i=[1,500]$ ). Set  $T=100$ ,  $\omega = 0.5$  and  $Tb_i=(0, 1, 0, 1 \dots)$ . Based on formula (6),  $N_j$  is the main factor affecting  $HP$  value, and the range of  $N_j$  values is set to  $[1, 200]$  and  $J$  is set to 100. Then,  $No_j$  is set to  $[1, 200]$  and  $\alpha = 0.1$ . In formula (7),  $V_y$  is set to  $[1, 500]$ .  $Y=100$  and  $T_y=1$ .  $\rho$  is set to  $(0, 1, 0, 1, \dots)$  and  $\mu$  is set to  $(1, 0, 1, 0, \dots)$  for each  $V_y$ . In formula (8),  $B_n=100$  and  $B_l$  is set to  $[1, 100]$ .  $S_r=500$  and  $S_b$  is set to  $[1, 10]$ .  $\lambda=\delta=1$  and  $D_n=D_e$ . Meanwhile, the training values ( $TV$ ) are approximated to ten integers from 1 to 10.

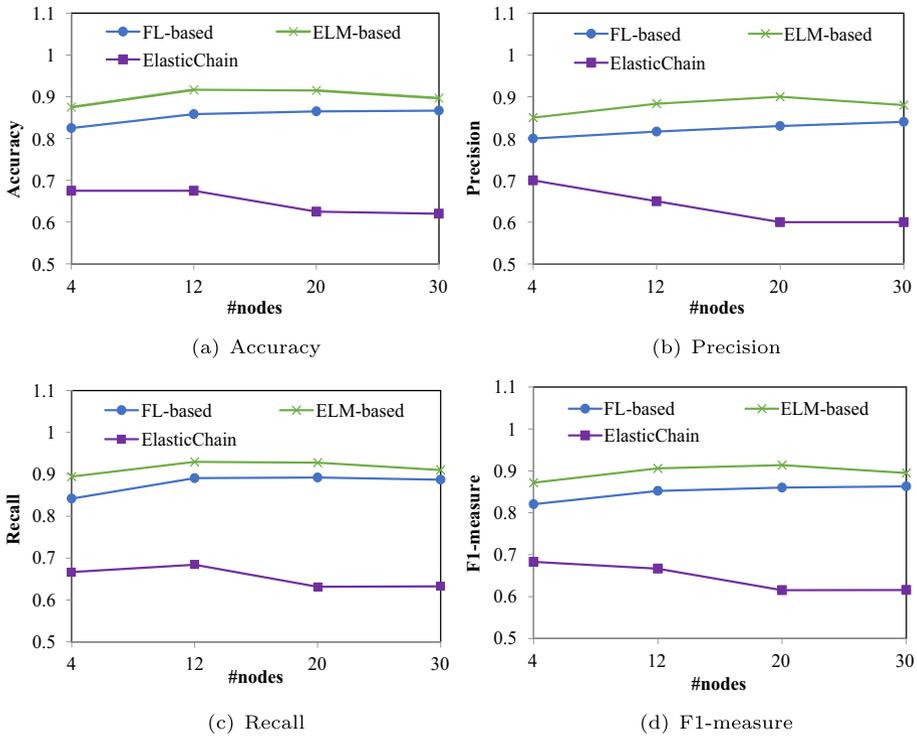


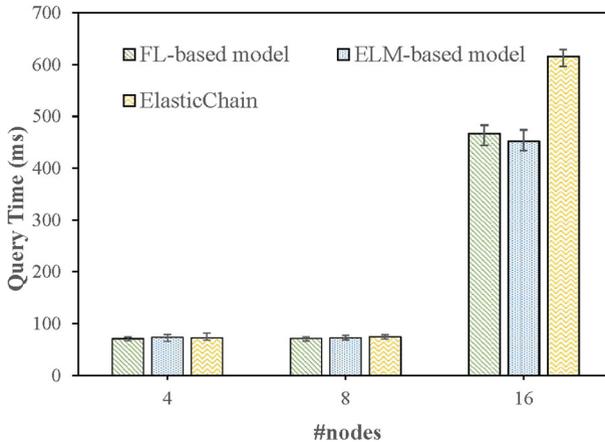
Figure 4 The classification results of different models

### 5.2 Evaluation of the FL-based classifier

In the experiment, the data set is divided into a training set with 1200 groups of features and a test set with 300 groups of features. To verify that the FL-based blockchain storage model has high classification performance. We compare our experimental results with existing ELM based methods, and the Fed-ELMS classifier in the storage model is replaced with ELM classifier. Then, the two classifiers are trained and tested respectively. The sigmoidal function is used in these models. The number of hidden layer nodes in the classifiers is 10, and the chunk size is set to 20.

We experimented on the accuracy, precision, recall and F1-measure of the classification results. The classification results of the FL-based blockchain storage model and OS-ELM-based model and the ElasticChain model are shown in Figure 4 when there are 4, 12, 20 and 30 nodes in these models.

We can get the following conclusions from Figure 4. In terms of accuracy, precision, recall and F1 measure, the FL-based blockchain storage model is smaller than the ELM-based model, but the gap is small. The reason is that the training based on ELM model is centralized, and all training data are saved in the local database. In the FL-based blockchain storage model, the training data is stored in different nodes, and the parameters need to be transmitted to the centralized ELM server after local training. Compared with the two models, the data features of the FL-based model may not be read completely, and the centrally saved data features are more comprehensive, so the experimental results of ELM-based model are



**Figure 5** The average query time for one transaction

slightly better. However, it can be seen from the figures that the FL-based model can still be well trained, but its accuracy is slightly reduced.

The accuracy, precision, recall and F1 measure values of FL-based blockchain storage model and ELM-based model are much higher than those of ElasticChain model. The reason is that the feature of the evaluation hot nodes in the ElasticChain model is insufficient. The reliability of the nodes is only taken into account when storing blocks in the ElasticChain model. FL-based blockchain storage model and ELM-based model evaluate hot nodes from five aspects of blocks, and efficient machine learning classification methods are adopted, so their classification results have obvious advantages.

The four values of accuracy, precision, recall and F1 measure of the FL-based blockchain storage model show a slight upward trend with the increase of the number of nodes and gradually become stable. The reason is that when the number of nodes is small, the feature data of nodes is small, and the training of models is insufficient. When the number of nodes increases, the number of training data also increases. The test results of the model improve and gradually stabilize.

### 5.3 Evaluation of the query performance

Then, the query performance of FL-based model, ELM-based model and ElasticChain model is tested and compared. The FL-based hot block storage strategy is adopted in FL-based model, and ELM classifier is used in the storage strategy of ELM-based model. FL-based model and ELM-based model are founded on the ElasticChain system, and the ElasticChain systems are implemented by Fabric v0.6. Fabric v0.6, as the initial implementation method of the alliance blockchain, has been widely used and studied.

In the three models, the *example02.go* [36] chaincode is deployed, and 5.39KB of data is generated and broadcasted after one transaction is confirmed. The block size is set to 100, and when 2000 transactions are confirmed, the 20 new blocks are saved in storage nodes according to the duplicate ratio regulation algorithm [6]. Randomly select 10 transactions for query, and record the query time. When 4, 8 and 16 nodes are established in the three systems, the time spent in querying a transaction is shown in Figure 5.

We can get the following conclusions from Figure 5.

When 16 nodes exist in the FL-based model, ELM-based model and ElasticChain model, the average query time of a transaction in the three systems is much higher than that when there are four or eight nodes in the systems. However, when there are 4 nodes in the 3 systems, the transaction query time is almost the same as that of the systems with 8 nodes. The reason is that the three systems are implemented based on the ElasticChain model. The duplicate ratio regulation algorithm is adopted in the ElasticChain models. In our experiment, the minimum number of copies of a block in the algorithm is set to 8. When the number of nodes in the systems is less than or equal to 8, each node stores the complete data. When a data query is executed, it will be performed in the local database. However, when the number of nodes in the systems is greater than 8, according to the duplicate ratio regulation algorithm, not all the block data is stored in each node. When a node queries a block data, the target data may not be stored locally, so it is necessary to issue a query request to other nodes and return the target data. Communication between nodes takes up a lot of the query time. As a result, the total query time increases.

When there are 16 nodes in the systems, the average query time of a transaction in the FL-based model and ELM-based model is lower than that in the ElasticChain model. The reason is that in the FL-based model and ELM-based model, more features of hot blocks are considered, and machine learning methods are adopted, so that hot blocks can be found more accurately and saved locally. Therefore, the query response time is reduced. Meanwhile, the average query time of a transaction in the FL-based model is slightly higher than that in the ELM-based model. The reason is that the classification accuracy of Fed-ELM classifier for hot blocks is slightly lower than that of ELM classifier. Even more remarkably, compared with the ELM-based model, the feature data of nodes in FL-based model is not sent to other nodes, and the privacy information of nodes is effectively protected.

## 6 Conclusion

In this paper, we proposed the FL-based blockchain storage model to achieve fast query under the premise of protecting data privacy in the Internet of Things environment, where the resource-constrained IoT edge nodes store part of blockchain blocks. In the proposed model, five features of hot blocks are defined: objective feature, historical popularity, potential popularity, storage requirements and training value, and the hot block storage strategy is designed based on these features. Finally, the experimental results on synthetic data demonstrate the accuracy and efficiency of the FL-based blockchain storage model.

In the future, IoT nodes in various scenarios will be added to the new model. There may be differences in the evaluation criteria of hot nodes under different scenarios. Some new features for evaluating hot nodes need to be added to the model. Meanwhile, the feature supplementation will not affect the framework of the FL-based blockchain storage model.

**Author Contributions** Dayu Jia: Writing-Original Draft; Guanghong Yang: Review and Editing; Min Huang: Funding acquisition; Junchang Xin: Review and Editing; Guoren Wang: Supervision; George Y. Yuan: Provide valuable suggestions in the review.

**Funding** This work is supported by the National Key Research and Development Program of China (Grant Nos. 2021YFB3300900, 2020YFE0201100 and 2022YFB4500800), the Funds of the National Natural Science Foundation of China (Grant Nos. 61621004, U1908213 and 62072089), the Research Fund of State Key Laboratory of Synthetical Automation for Process Industries (Grant No. 2018ZCX03), the Key Scientific Research Project of Liaoning Provincial Department of Education (Grant No. LZD202002), the Fundamental Research Funds for the Central Universities (Nos. N2116016, N2104001 and N2019007), the Open Program of Neusoft Corporation (No. NCBETOP2102), Program (No. JCKY2021211B017).

**Data Availability** The data that support the findings of this study are available on request from the corresponding author, upon reasonable request.

## Declarations

**Competing interest** The authors declare that they have no conflict of interest.

**Ethical approval** All applicable international, national, and/or institutional guidelines for the care and use of animals were followed.

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

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

## References

1. Li, H., Tang, B., Lu, H., Cheema, M.A., Jensen, C.S.: Spatial data quality in the iot era: management and exploitation. SIGMOD Conference, 2474–2482 (2022)
2. Ni, W., Cheng, P., Chen, L.: Mixing transactions with arbitrary values on blockchains. IEEE International Conference on Data Engineering (ICDE), 2602–2614 (2022)
3. Anuradha, M.P., Rani, K.L.F.C.: Chapter fourteen - blockchain technology for iot edge devices and data security. Adv. Comput. **127**, 379–412 (2022)
4. Song, Q., Chen, Y., Zhong, Y., Lan, K., Fong, S., Tang, R.: A supply-chain system framework based on internet of things using blockchain technology. ACM Trans. Internet Techn. **21**(1), 1–24 (2021)
5. Bhattacharya, P., Tanwar, S., Bodkhe, U., Tyagi, S., Kumar, N.: Bindaas: blockchain-based deep-learning as-a-service in healthcare 4.0 applications. IEEE Trans. Netw. Sci. Eng. **8**(2), 1242–1255 (2021)
6. Jia, D., Xin, J., Wang, Z., Guo, W., Wang, G.: Elasticchain: Support very large blockchain by reducing data redundancy. Proc. APWeb-WAIM, 440–454 (2018)
7. Jia, D., Xin, J., Wang, Z., Lei, H., Wang, G.: Elm-based data distribution model in elasticchain. World Wide Web **25**(3), 1085–1102 (2022)
8. Jia, D., Xin, J., Wang, Z., Wang, G.: Optimized data storage method for sharding-based blockchain. IEEE Access **9**, 67890–67900 (2021)
9. Dang, H., Dinh, T.T.A., Lohin, D., Chang, E.C., Lin, Q., Ooi, B.C.: Towards scaling blockchain systems via sharding. Proc. SIGMOD, 123–140 (2019)
10. Kogias, E.K., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., Ford, B.: Omniledger: a secure, scale-out, decentralized ledger via sharding. Proc. IEEE Symposium on Security and Privacy, 583–598 (2018)
11. Huang, G.-B., Siew, C.K.: Extreme learning machine: Rbf network case. Int. Conference on Control, Automation, Robotics and Vision, ICARCV, 1029–1036 (2004)
12. Shahverdi, A., Shirinov, M., Dachman-Soled, D.: Database reconstruction from noisy volumes: a cache side-channel attack on sqlite. USENIX Security Symposium, 1019–1035 (2021)
13. Jia, J., Gong, N.Z.: Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. USENIX Security Symposium, 513–529 (2018)
14. Kshetri, N.: Web 3.0 and the metaverse shaping organizations’ brand and product strategies. IT Prof. **24**(2), 11–15 (2022)
15. Leite, L., dos Santos, D.R., Almeida, F.L.: The impact of general data protection regulation on software engineering practices. Inf. Comput. Secur. **30**(1), 79–96 (2022)
16. Stallings, W.: Handling of personal information and deidentified, aggregated, and pseudonymized information under the california consumer privacy act. IEEE Secur. Priv. **18**(1), 61–64 (2020)
17. Dai, H.-N., Wu, Y., Wang, H., Imran, M., Haider, N.: Blockchain-empowered edge intelligence for internet of medical things against covid-19. IEEE Internet Things Mag. **4**(2), 34–39 (2021)
18. Kumar, P., Kumar, R., Gupta, G.P., Tripathi, R., Srivastava, G.: P2tif: a blockchain and deep learning framework for privacy-preserved threat intelligence in industrial iot. IEEE Trans. Ind. Informatics **18**(9), 6358–6367 (2022)
19. Kumar, P., Kumar, R., Srivastava, G., Gupta, G.P., Tripathi, R., Gadekallu, T.R., Xiong, N.N.: Ppsf: a privacy-preserving and secure framework using blockchain-based machine-learning for iot-driven smart cities. IEEE Trans. Netw. Sci. Eng. **8**(3), 2326–2341 (2021)

20. Jamil, F., Ibrahim, M., Ullah, I., Kim, S., Kahng, H.K., Kim, D.: Optimal smart contract for autonomous greenhouse environment based on iot blockchain network in agriculture. *Comput. Electron. Agric.* **192**, 106573 (2022)
21. Bedari, A., Wang, S., Yang, J.: A two-stage feature transformation-based fingerprint authentication system for privacy protection in iot. *IEEE Trans. Ind. Informatics* **18**(4), 2745–2752 (2022)
22. Akter, M., Moustafa, N., Lynar, T.M.: Edge intelligence-based privacy protection framework for iot-based smart healthcare systems. *INFOCOM Workshops*, 1–8 (2022)
23. Guan, Z., Yang, W., Zhu, L., Wu, L., Wang, R.: Achieving adaptively secure data access control with privacy protection for lightweight iot devices. *Sci. China Inf. Sci.* **64**(6), 140–153 (2021)
24. He, X., Guo, H., Cheng, X.: Blockchain-based privacy protection scheme for iot-assisted educational big data management. *Wirel. Commun. Mob. Comput.* **3558972**, 1–11 (2021)
25. Zhang, Z., Dong, D., Ma, Y., Ying, Y., Jiang, D., Chen, K., Shou, L., Chen, G.: Refiner: a reliable incentive-driven federated learning system powered by blockchain. *Proc. VLDB Endow* **14**(12), 2659–2662 (2021)
26. Deng, Y., Han, T., Zhang, N.: Flex: Trading edge computing resources for federated learning via blockchain. *INFOCOM Workshops*, 1–2 (2021)
27. Feng, L., Zhao, Y., Guo, S., Qiu, X., Li, W., Yu, P.: BafI: a blockchain-based asynchronous federated learning framework. *IEEE Trans. Computers* **71**(5), 1092–1103 (2022)
28. Qin, Z., Ye, J., Meng, J., Lu, B., Wang, L.: Privacy-preserving blockchain-based federated learning for marine internet of things. *IEEE Trans. Comput. Soc. Syst.* **9**(1), 159–173 (2022)
29. Huang, G.-B., Zhu, Q.-Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**(1–3), 489–501 (2006)
30. Ouyang, T., Wang, C., Yu, Z., Stach, R., Mizaikoff, B., Huang, G.-B., Wang, Q.-J.: Nox measurements in vehicle exhaust using advanced deep elm networks. *IEEE Trans. Instrum. Meas.* **70**, 1–10 (2021)
31. Chen, Y.-T., Chuang, Y.-C., Wu, A.-Y.A.: Online extreme learning machine design for the application of federated learning. *IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 188–192 (2020)
32. Zhang, Z., Cao, X., Jia, J., Gong, N.Z.: Fldetector: defending federated learning against model poisoning attacks via detecting malicious clients. *ACM SIGKDD Conference*, 2545–2555 (2022)
33. Donahue, K., Kleinberg, J.M.: Model-sharing games: analyzing federated learning under voluntary participations. *AAAI* **2021**, 5303–5311 (2021)
34. Zhang, J., Wu, Y., Pan, R.: Incentive mechanism for horizontal federated learning based on reputation and reverse auction. *WWW* **2021**, 947–956 (2021)
35. Bitcoin price stats and information. <https://bitinfocharts.com/>. [Online] Accessed on: 23 Sep 2022
36. Chaincode. [https://github.com/hyperledger/fabric/blob/v0.6/examples/chaincode/go/chaincode\\_example02/chaincode\\_example02.go](https://github.com/hyperledger/fabric/blob/v0.6/examples/chaincode/go/chaincode_example02/chaincode_example02.go). [Online] Accessed on: 23 Sep 2022

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Dayu Jia<sup>1</sup> · Guanghong Yang<sup>1,2</sup> · Min Huang<sup>1,2</sup> · Junchang Xin<sup>3,4,5</sup> · Guoren Wang<sup>6</sup> · George Y. Yuan<sup>6,7</sup>

Guanghong Yang  
yangguanghong@ise.neu.edu.cn

Min Huang  
mhuang@mail.neu.edu.cn

Junchang Xin  
xinjunchang@mail.neu.edu.cn

Guoren Wang  
wanggr@bit.edu.cn

George Y. Yuan  
yuanye@thinvent.com

- <sup>1</sup> College of Information Science and Engineering, Northeastern University, 110819 Shenyang, China
- <sup>2</sup> State Key Laboratory of Synthetical Automation of Process Industries, Northeastern University, 110819 Shenyang, China
- <sup>3</sup> School of Computer Science and Engineering, Northeastern University, 110819 Shenyang, China
- <sup>4</sup> Key Laboratory of Big Data Management and Analytics, Northeastern University, 110819 Shenyang, China
- <sup>5</sup> Neusoft Corporation (Research Center of Liaoning Promotion for Blockchain Engineering Technology), Shenyang, China
- <sup>6</sup> School of Computer Science and Technology, Beijing Institute of Technology, 100081 Beijing, China
- <sup>7</sup> Thinvent Digital Technology Co.,Ltd., Nanchang, China