IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 26, 2021, accepted May 2, 2021, date of publication May 5, 2021, date of current version May 13, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3077650*

# Optimized Data Storage Method for Sharding-Based Blockchain

**DAYU JIA**[1], **JUNCHANG XIN**[1,2], **ZHIQIONG WANG**[3,4], **AND GUOREN WANG**[5]
[1]School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
[2]Key Laboratory of Big Data Management and Analytics, Northeastern University, Shenyang 110169, China
[3]College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110169, China
[4]Neusoft Research of Intelligent Healthcare Technology Company Ltd., Shenyang 110167, China
[5]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Junchang Xin (xinjunchang@mail.neu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62072089; and in part by the Fundamental Research Funds for the Central Universities under Grant N2116016, Grant N161602003, Grant N180408019, and Grant N180101028.

**ABSTRACT** COVID-19 virus is raging across the planet. In countries where the epidemic is under control, the main mode of virus transmission is through the transport of imported refrigerated food from epidemic areas. Blockchain is a great way for the government to trace every piece of food. However, the high-performance requirements of the blockchain system for nodes limit its wide application. Several sharding-based blockchain systems have been proposed to solve this limitation. Which blocks should be saved by nodes in the sharding-based blockchain system is a new problem. To solve this problem, the optimized data storage method is proposed in this paper. Five features of block popularity are presented, including the objective feature of a block, the objective feature of the block associated with the node, the historical popularity, the hidden popularity and the storage requirements. Then the ELM classifier is used in the optimized model due to its high performance of training and classification. Finally, the experimental results on synthetic data demonstrate the accuracy and efficiency of the optimized data storage model.

**INDEX TERMS** Blockchain, hot block, classification, sharding technology, extreme learning machine.

## I. INTRODUCTION

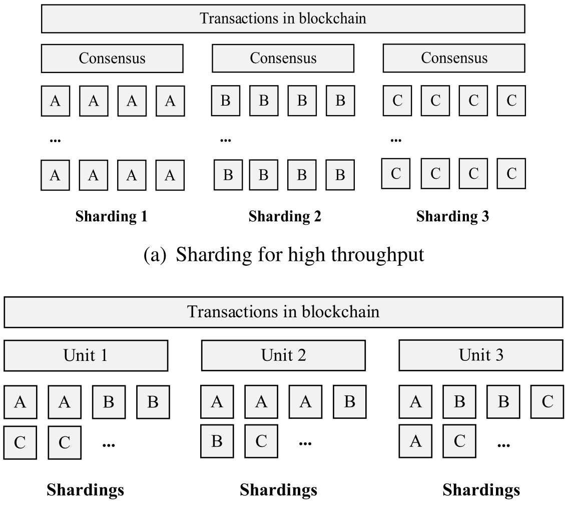
Blockchain and artificial intelligence are regarded as the two innovative technologies that are most likely to increase the productivity of human society in the next ten years [1]. Moreover, these two technologies, together with cloud computing and data science, are collectively referred to as "ABCD", which is the future direction of information technology research. At present, blockchain technology has been widely used in many fields such as finance, database, medical treatment and government work.

However, the original blockchain technology has two pain points. One is that full nodes have to jointly maintain the same ledger, which results in a low throughput that cannot meet the needs of real-world applications such as banking transactions. The other is that the blockchain system requires each full node to keep a complete copy of the blockchain, which severely limits the joining of resource-constrained nodes. In order to solve these two pain points, there has been a lot of research. Sharding technology is one of the widely used methods [2]–[7].

The current sharding technology mainly has two structures, as shown in Figure 1. One structure is that a group of resource-constrained nodes form an organization to complete all tasks of a transaction. An organization stores the same consensus transaction data, but each organization stores different shard data, such as OmniLedger [2], RapidChain [3], Elastico [4] and another sharding chain [5]. Another structure, such as Consensus Unit [6] and Elasticchain [7], where some resource-constrained nodes form a whole to complete the work of a full node in the blockchain system. These nodes reach a consensus with other full nodes or composed full nodes. Each of the full nodes (or composed full nodes) stores the same data.

Both structures have their own advantages. The first structure can achieve visa-level throughput. While transaction data in the second structure is more secure because it is stored and maintained by all full nodes. The research work in this paper is based on the second sharding structure.

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.



(a) Sharding for high throughput

(b) Sharding for more nodes

**FIGURE 1. Sharding technology in blockchian.**

In the application of sharding-based blockchain, each resource-constrained node only saves part of the blockchain data due to the huge amount of data in a ledger. When a node needs to read data, it needs to initiate a query request to other nodes if the data is not stored locally. However, in the current sharding methods, no effective method has been proposed to tell nodes which sharding data should be saved. For example, (details in related work) in [7], under the premise of ensuring that the total number of shards in the system is sufficient, the resource-constrained node randomly saves the shard data. [6] proposed a method for nodes to save sharding data, but it considers fewer parameters, only including the storage space of nodes and the frequency of node access data. In the current blockchain system, if a node needs to access other nodes when reading data, in some scenarios with poor network environments and frequent queries, the query efficiency will be seriously affected.

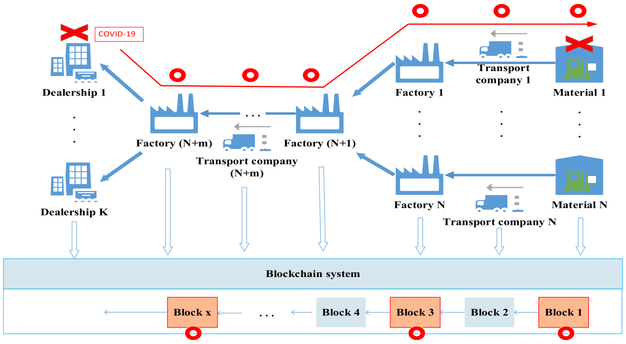


**FIGURE 2. A scenario for supply chain.**

Take the application scenario of the supply chain as an example. As shown in Figure 2, the supply chain has 2 characteristics: (1) There are a large number of independent participants, including cross-country production factories, logistics service providers and Dealerships. (2) The relationship between participants is close. Any change in one link will cause fluctuations in other links, and the impact will be amplified step by step.

According to the characteristics of the supply chain, its current pain points are as follows: (1) The cost of information exchange between enterprises is high because data sharing is hindered by data islands and privacy protection. (2) Commodity traceability and anti-counterfeiting are difficult to achieve because there is no guarantee that the data provided by a party in the supply chain is absolutely true and reliable.

Pain point 2 is highlighted when the COVID-19 virus is ravaging the world. In some countries with better epidemic prevention work, such as China, the main mode of virus transmission is to infect and spread the virus when transporting imported refrigerated food from epidemic areas. At present, China's cold chain (a kind of supply chain application) information system is imperfect, and a lot of resources are wasted to search for the source of the virus. Therefore, the pain point of the supply chain not only spawns huge economic losses but also affects people's health.

The combination of blockchain and Internet of Things (IoT) technology can solve the current pain points of the supply chain. The data detected by edge devices in the supply chain are stored in the blockchain through IoT technology. Blockchain, through privacy protection mechanisms such as information encryption and decryption, and zero-knowledge proof, can remove the obstacles caused by data privacy to data sharing. At the same time, the data structure constructed by the blockchain in a peer-to-peer network environment has the characteristics of data traceability and anti-forgery.

Some blockchain (BC) systems applied to the IoT scenarios have been proposed, such as an optimized blockchain (BC) [21], which employs a hierarchical architecture that uses a centralized private Immutable Ledger (IL) at the local IoT network level to reduce overhead, and a decentralized public BC at higher end devices for stronger trust. The optimized BC eliminates the overhead associated with the classic blockchain while retaining its security and privacy benefits.

As shown the red line in Figure 2, the government can quickly trace all locations and persons (red circles) related to the virus based on the relevant data in the blockchain (red blocks).

Moreover, most of the edge devices of the IoT are resource-constrained nodes and cannot be added to the blockchain system as full nodes. Therefore, the blockchain system using sharding technology is suitable for supply chains.

In some scenarios in the supply chain, the edge device is in a poor network environment, and the query demand is high. For example, in cold chain transportation, trucks are often located in remote areas, and the devices in the cold box detect the temperature of the food in real time. Truck drivers, shippers, and consignees all need to frequently check the temperature of food. If the query cannot be responded to in time, the state of the food temperature greater than zero degrees Celsius will not be discovered in time.

The consignees cannot quickly take countermeasures and cause additional losses.

Therefore, it is necessary to improve the query efficiency of the sharding-based blockchain. This paper first proposes an optimized data storage model based on blockchain sharding technology. For each node in the optimization model, blocks can be divided into hot blocks and non-hot blocks by the Extreme Learning Machine (ELM) method. After classification, each node saves the most relevant hot block. When a node initiates a query, it can query locally instead of frequently sending query requests to other nodes.

Then, we define a hot block for a node. A hot block is comprehensively evaluated from the 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 in the system, so that the node can accurately find the most relevant hot block and store it locally.

Specifically, the major contributions of this paper are the followings:

- We propose an optimized data storage model based on blockchain sharding technology. The ELM method is used in this model to be the classifier in order to improve classification efficiency.
- We design an evaluation method of hot block for a node. According to this evaluation standard, nodes can classify the most relevant hot block and store it locally.
- We conduct a set of experiments to demonstrate the accuracy of the optimized data storage model and the query efficiency of the blockchain system using the new model based on the synthetic data.

The remainder of the paper is organized as follows. Section 2 reviews the related work on the blockchain technology in IoT and the sharding-based blockchain technologies. Section 3 introduces the background of ELM. Section 4 introduces the architecture of the optimized data storage model and the strategies of feature selection. Section 5 reports experimental evaluation. Finally, conclusions are presented in Section 6.

## II. RELATED WORK

### A. BLOCKCHAIN SOLUTIONS FOR IoT

Currently, many researchers are working on the blockchain application in IoT. Such as, [8] proposes a blockchain-based framework for data integrity service without relying on any Third Party Auditor. Reference [9] proposes CreditCoin, a blockchain-based announcement network, which implements a reliable vehicular announcement by a user who does not reveal identity. Reference [10] implements a fully distributed access control system based on blockchain to manage billions of IoT devices in a unified manner. This system frees up a large of space and performance of edge devices. Reference [22] proposes a Memory Optimized and Flexible BC (MOF-BC) that enables the IoT users and service providers to remove or summarize their transactions and age their data. MOF-BC introduces the notion of a Generator Verifier (GV) which decreases BC memory consumption effectively. Reference [23] explores key benefits and design challenges for blockchain technologies, and potential applications of blockchain technologies for IoT. One of the most important challenges is the scalability of blockchain technologies does not meet the IoT application requirements.

### B. SHARDING-BASED BLOCKCHAIN SYSTEMS

Consensus Unit [6] is proposed to address the high storage requirement in the wide usage of blockchain on various devices such as mobile phones or low-end PCs. A Consensus Unit organizes different nodes into one unit and lets them store at least one copy of blockchain data in the system together. Based on this structure, [6] proposed a block allocation method to make full use of storage space and minimize query costs. The definition of query cost only considers how often the block is queried. However, there are many features that affect the relevance of a block to this node, such as the number of transactions contained in the block, the transaction value recorded in the block, the number of transactions related to the node in the block, etc. In this paper, the query cost is evaluated more comprehensively and accurately. According to the evaluation results, more relevant data will be stored in the node, and query response time will be reduced.

Elasticchain [7] is another sharding-based blockchain system. Nodes in ElasticChain store the shardings of the complete chain based on the duplicate ratio regulation algorithm. Meanwhile, the node reliability verification method was used for increasing the stability of full nodes and reducing the risk of data imperfect recovering caused by the reduction of duplicate numbers. However, data security is only considered in the duplicate ratio regulation algorithm. The algorithm gives the minimum number of copies stored for each block sharding, and nodes will be randomly selected to store these shardings. In this case, the stored data in the node is very likely to be irrelevant to itself. When the node initiates a query request, more time will be spent to accessing other nodes. The Optimized data storage method proposed in this paper will alleviate this problem.

## III. PRELIMINARIES

In this paper, ELM will be used as a classifier to distinguish whether a block is a hot spot of nodes. In this section, we give some preliminaries of this work including the theory and advantage of ELM, then we propose the problem definition.

### A. THE THEORY OF ELM

ELM is originally developed for single hidden-layer feedforward neural networks (SLFNs) and then extended to the "generalized" SLFNs where the hidden layer need not be neuron alike [11], [12]. ELM first randomly assigns the input weights and the hidden layer biases, and then analytically determines the output weights of SLFNs. ELM can achieve better generalization performance than other conventional learning algorithms at an extremely fast learning speed. Besides, ELM is less sensitive to user-specified parameters and can be deployed faster and more conveniently [13].

The theory of ELM is as follows [14]. For $n$ arbitrary distinct samples $(x_j, t_j)$, where $x_j = [x_{j1}, x_{j2}, \ldots, x_{jn}]^T \in \mathbf{R}^n$,and $t_j = [t_{j1}, t_{j2}, \ldots, t_{jm}]^T \in \mathbf{R}^m$, standard SLFNs with $L$ hidden nodes and activation function $g(x)$ are mathematically modeled as

$$\sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta_i g(w_i \cdot x_j + b_i) = o_j \quad (j = 1, 2, \ldots, N) \tag{1}$$

where $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is the weight vector connecting the $i$th hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the weight vector connecting the $i$th hidden node and the output nodes, $b_j$ is the threshold of the $i$th hidden node, and $o_j = [o_{j1}, o_{j2}, \ldots, o_{jm}]^T$ is the $j$th output vector of the SLFNs.

The standard SLFNs with $L$ hidden nodes and activation function $g(x)$ can approximate these $N$ samples with zero error. It means $\sum_{j=1}^{L} ||o_j - t_j|| = 0$ and there exist $\beta_i$, $w_i$ and $b_i$ such that

$$\sum_{i=1}^{L} \beta_i g(w_i \cdot x_j + b_i) = t_j \quad (j = 1, 2, \ldots, N) \tag{2}$$

The equation above can be expressed compactly as follows.

$$H\beta = T \tag{3}$$

where

$$H(w_1, \ldots, w_L, b_1, \ldots, b_L, x_1, \ldots, x_L) = [h_{ij}] = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \ldots & g(w_L \cdot x_1 + b_L) \\ g(w_1 \cdot x_2 + b_1) & \ldots & g(w_L \cdot x_2 + b_L) \\ \vdots & \vdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \ldots & g(w_L \cdot x_N + b_L) \end{bmatrix}_{N \times L} \tag{4}$$

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{L1} & \beta_{L2} & \cdots & \beta_{Lm} \end{bmatrix}_{L \times m} \tag{5}$$

$$T = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1m} \\ t_{21} & t_{22} & \cdots & t_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ t_{N1} & t_{N2} & \cdots & t_{Nm} \end{bmatrix}_{N \times m} \tag{6}$$

$H$ is called the hidden layer output matrix of the neural network and the $i$th column of $H$ is the $i$th hidden node output with respect to inputs $x_1, x_2, \ldots, x_N$. The smallest norm least-squares solution of the above linear system is computed by

$$\beta = H^{\dagger} T \tag{7}$$

where $H^{\dagger}$ is the Moore-Penrose generalized the inverse of matrix $H$. Then the output function of ELM can be modeled as follows.

$$f(x) = h(x)\beta = h(x)H^{\dagger}T \tag{8}$$

Given a training set $\mathcal{N} = \{ (x_j, t_j) \mid x_j \in \mathbf{R}^n, t_j \in \mathbf{R}^m, j = 1, 2, \ldots, N \}$, activation function $g(w_i, b_i, x_j)$ and hidden node number $L$, the pseudo code of ELM [13] is given in Algorithm 1.

**Algorithm 1** ELM

```
for $i = 1$ to $L$ do
    Randomly assign input weight $w_i$
    Randomly assign input bias $b_i$
Calculate $H$
Calculate the output weight $\beta = H^{\dagger} T$
```

### B. THE ADVANTAGE OF ELM

ELM [14] is one of the machine learning models. We can produce reliable, repeatable decisions and uncover hidden insights through learning from historical relationships and trends in the data by using machine learning methods.

Compared to other machine learning methods, ELM has two advantages. First, ELM classifiers have a higher performance of training and classification [13]. For example [15], the testing accuracy of ELM is 99.14% in the MNIST OCR dataset, which is 0.27%, 0.09%, 0.54% and 0.42% higher than Deep Belief Networks (DBN), Deep Boltzmann Machines (DBM), Stacked Auto Encoders (SAE) and Stacked Denoising Auto Encoders (SDAE), respectively. The training time of ELM is 281.37s, while the training time of the other methods is more than 17 hours.

Second, online sequential learning can be achieved in ELM [16]. When new data is generated, traditional models need to integrate new data with old data and retrain. ELM can retain previous training experience and train new data based on current experience. Fast training can ensure that the training data of the model is complete and real-time.

Moreover, different from Deep Learning which requires intensive tuning in multiple hidden layers and hidden neurons, ELM theories show that hidden neurons are important but need not be turned for both SLFNs and multi-hidden-layer of networks [17]. The learning in ELM can simply be made without iteratively tuning hidden neurons.

Therefore, we use ELM as the classifier in this paper.

### C. PROBLEM DEFINITION

In the current sharding-based blockchain system, the evaluation of the correlation between a block and a node is not sufficient. This will cause the sharding data stored by the node to be irrelevant data of the node, which reduces the query efficiency.

#### 1) THREAT MODEL

A block may have the following characteristics: the transactions in the block contain a large number of users, a large

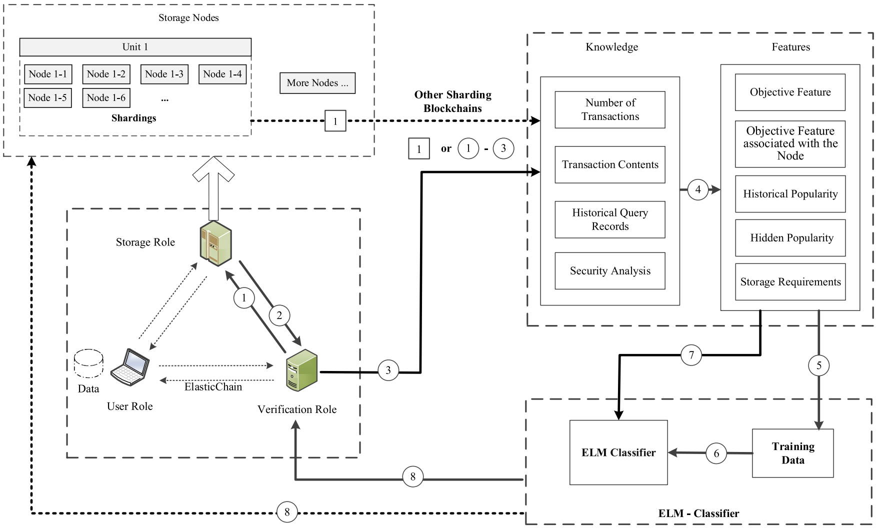


**FIGURE 3.** **Architecture of the optimized data storage model.**

number of transactions in the block are related to a node, and the block includes a large number of transactions of other nodes that are closely related to the node, etc.

According to the current evaluation method for the correlation between a block and a node, there are two situations that may happen: Some blocks are currently accessed frequently by a node, but may be rarely accessed in the future. The other is that the current node does not have high query requirements for a certain block, but it may need to visit this block frequently in future applications. When the above two situations occur, the blockchain system will frequently update the locally stored fragmented data, causing query delays.

Therefore, we propose an optimized data storage model in this paper, which can comprehensively describe the correlation between blocks and nodes, and accurately classify the hot blocks for a node.

## IV. THE OPTIMIZED DATA STORAGE MODEL

In this section, we first describe the architecture of the optimized data storage model. Then we introduce the features used to classify hot nodes. After that, we propose an algorithm to describe the data distribution process.

### A. ARCHITECTURE

Figure 3 shows the architecture of the optimized data storage model based on ELM. It mainly consists of three modules: the left is the sharding-based blockchain module; the upper right is the feature extraction module and the lower right is the classifier module.

In the sharding-based blockchain module, the lower part is the ElasticChain model. There are three roles for nodes [7]: the user node, the storage node, and the verification node. User nodes are participants in the blockchain system. Blockchain operations, such as transactions, are completed between user nodes. And the sharding blockchain data is stored in storage nodes. The verification node is to provide reliable storage nodes for the user nodes. The verification nodes visit and check the reliability of storage nodes at every same period time, and the two inspection results are returned, which are the integrity of the data in the storage nodes and number of successful verification by storage nodes.

In our optimized data storage model, the verification nodes also read the historical query records of the storage node. Meanwhile, the number of transactions, transaction contents in a block and the security of the block will be detected by verification nodes, as shown in the feature extraction module. Then, five features (objective feature of a block, objective feature of the block associated with the node, historical popularity, the hidden popularity and the storage requirements) will be evaluated based on the four pieces of knowledge. These features can describe the popularity of a block for a node completely.

The upper part of sharding-based blockchain module is the structure of storage nodes of ElasticChain. While the structure of other sharding-based blockchain systems like Consensus Unit [6] is like the upper part. The sharding data is uniformly distributed by the system. Therefore, the classification process is directly completed by the system.

Finally, in the classifier module, the hot blocks of a node are classified based on these five important features by using ELM. Then, nodes store the hot blocks as their sharding data. In ELM classifier, some of the blocks are sampled as training data. The sample blocks are the input of the classifier module. They consist of two kinds of blocks, hot blocks, and non-hot blocks. The way to create and sample training data are introduced in Section 4.3.

Next, we describe the feature selection process in detail.

### B. FEATURE SELECTION

In sharding-based blockchain systems, multiple features will affect the popularity of a block for a node, and we choose five important features among them in this paper. The chosen features are the objective feature of a block ($OF$), the objective feature of the block associated with the node ($OFN$), the historical popularity ($HIS$), the hidden popularity ($HID$) and the storage requirements ($SR$). Correct evaluations on the popularity of a block can be made in most cases by using these five features.

Admittedly, other features may also affect the popularity of blocks in some special scenarios. The features can be added to the feature extraction module without changing the structure of optimized model.

The five features we proposed are all calculated from other two to four features. We input the five summarized features into the ELM classifier because we want to reduce the dimension of classifier and increase the speed of classification.

The blocks $B$ ($B = \{b_1, b_2, \ldots, b_I\}$) will be detected at a fixed interval. The five features are updated after each detection. When $I$ blocks are produced, $I$ sets of feature data will be generated. Each set has five features, so the data sets ($DS$) of blocks can be expressed as a 5 × $I$ matrix:

$$DS = \begin{bmatrix} OF_1 & OFN_1 & HIS_1 & HID_1 & SR_1 \\ OF_2 & OFN_2 & HIS_2 & HID_2 & SR_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ OF_I & OFN_I & HIS_I & HID_I & SR_I \end{bmatrix}_{5\times I} \tag{9}$$

Then, we define five block popularity features for a node.

*Definition 1 (Objective Feature of a Block (OF)):* The objective feature is the evaluation of a block based on its basic characteristics. A block ($b_i$ ($1 \leq i \leq I$)) consists of a block header ($h_i$) and a block body ($d_i$), i.e. $b_i = \{h_i, d_i\}$. Many transactions ($t$) exist in a block body ($d_i = \{t_1, t_2, \ldots, t_j\}$). A transaction $t_j$ includes the initiator of the transaction ($IT_j$), the receiver of the transaction ($RT_j$) and the transaction value ($TV_j$), i.e. $t_j = \{IT_j, RT_j, TV_j\}$.

First, let the number of transactions in a block be $N_t$, and the total number of users involved in the transactions be $N_u$ ($N_u$ is the sum of $IT$ and $RT$). If the $N_t$ and $N_u$ of a block are large, the block is likely to be accessed and queried by users in the future. Then, the greater the total value ($TV$) of all transactions in a block ($TV$ is the sum of each $TV_j$), the more important the block is. Moreover, the location of the block will also affect the objective popularity feature of a block. An old data may be accessed less often than new data in most cases.

The objective feature of a block ($OF$) can be expressed as follows:

$$OF = (N_t + N_u) \times \sum_{k=1}^{j}(TV_k) \times a^{-(I-i)} \tag{10}$$

where $a^{-(I-i)}$ is the coefficient of block location ($a > 1$). $I$ is the number of the lastest block, and $i$ ($1 \leq i \leq I$) is the number of the evaluated block. Therefore, the $OF$ of a block will decrease the addition of blocks.

*Definition 2 (Objective Feature of the Block Associated With the Node (OFN)):* The objective feature of a block ($OF$) can make a unified evaluation for all blocks, and the evaluation is macroscopic. In the sharded blockchain system, each node keeps a number of blocks in which the data is commonly used by the nodes. Therefore, it is also necessary to objectively evaluate the block from a micro perspective, which is related to the node. If the value of the objective feature of a block is not very large, but the features are strongly correlated with a node, the node will also have a high probability to query the data in this block in the future.

The objective feature of a block associated with the node ($OFN$) can be expressed as follows:

$$OFN = N_n \times \sum_{k=1}^{j}(TVn_k) \times b^{-(I-i)} \tag{11}$$

Here, when a node store a block $b_i$, $N_n$ is the number of transactions which the initiator or receiver is this node in this block. $TVn$ is the total value of these $N_n$ transactions. $b^{-(I-i)}$ is the coefficient of block location ($b > 1$), which is the same as in formula (10).

*Definition 3 (Historical Popularity (HIS)):* Besides the objective features of blocks, the historical record of a block read by nodes also has a greater impact on the block popularity. By analyzing historical query records, we evaluate the historical popularity of a block for a node from three aspects: the usage frequency of the block, the time since the block is used, and the response time of each query. When a block is frequently used by a node, the node should store the block locally to reduce the time cost in the next query. However, if some blocks have not too many total visits, but they have been visited many times recently, these blocks can also be regarded as hot blocks. Moreover, if the query response time of some blocks is long, it means that the node that saves this block is in a poor network environment. We also need to consider storing the blocks with slower query response locally. The historical popularity ($HIS$) can be expressed as follows:

$$HIS = \sum_{p=1}^{P}\left(\frac{1}{TI_p \times QT_p}\right) \tag{12}$$

Here, $P$ is the total number of times the block has been accessed. $p$ ($1 \leq p \leq P$) is the $p^{th}$ access. $TI_p$ is the time interval between the $p^{th}$ access and the current. $QT_p$ is the query time spent in the $p^{th}$ access.

*Definition 4 (Hidden Popularity (HID):* Node $A$ stores the block $b_i$, and $b_i$ records many transactions $\{t_1, t_2, \ldots, t_j\}$, which include the initiators of the transaction ($\{IT_1, \ldots, IT_j\}$), the receivers of the transaction ($\{RT_1, \ldots, RT_j\}$) and the transaction values ($\{TV_1, \ldots, TV_j\}$). In some case, the objective feature value and historical popularity value of this block are not very large due to the small transaction values ($\{TV_1, \ldots, TV_j\}$). However, if the transaction data of these nodes ($\{IT_1, \ldots, IT_j\}$ and $\{RT_1, \ldots, RT_j\}$) stored in other blocks is accessed frequently in the recent past by node $A$, although block $b_i$ is not a hot block currently, it has great potential to become a hot block in the future. The reason is that node $A$ is likely to access the data related to these nodes ($\{IT_1, \ldots, IT_j\}$ and $\{RT_1, \ldots, RT_j\}$) in block $b_i$. We define the hidden popularity of a block to evaluate the potential of the block to become a hot block. The hidden popularity of a block can be evaluated by formula 13.

$$HID = \sum_{q=1}^{Q} DT_q \times (\eta + \xi) \tag{13}$$

Here, we first set a fixed time $T_f$, and the queries within time $T_f$ is considered to be the recent queries. If the processing power is limited or the hidden popularity needs to be obtained quickly, we can reduce the setting of $T_f$. There are $Q$ transactions in the recent time $T_f$. $DT_q$ is the distance between the time when the $q^{th}$ transaction was queried and the present. If the initiator of the $q^{th}$ $(1 \leq q \leq Q)$ transaction also has a related transaction in block $b_i$, then $\eta = 1$, otherwise $\eta = 0$. Similarly, if the receiver of the $q^{th}$ transaction also has a related transaction in block $b_i$, then $\xi = 1$, otherwise $\xi = 0$.

*Definition 5 (Storage Requirements (SR)):* Most sharding-based blockchain systems have a minimum requirement for the number of copies of a block to be stored. For example, in [6], the system requires that each Consensus Unit need to save a complete blockchain data. Therefore, as a member of the Consensus Unit, a node cannot only store blocks with high objective feature values. The nodes in the Consensus Unit must meet the requirements of the system. As another example, Elasticchain [7] proposes the Duplicate Ratio Regulation algorithm, which analyzes the security of each block and sets the minimum number of copies stored in each block. The number of blocks stored by the nodes in Elasticchain system needs to reach this minimum value, and the nodes cannot blindly save blocks with higher popularity.

Therefore, we define storage requirements ($SR$) to describe this feature of blocks. For example, for blocks whose number of copies does not meet the requirements of the sharding-based blockchain system, the value of the storage requirements for these blocks will increase, and the popularity of this block will increase. In this way, nodes in the blockchain systems will store these blocks locally and meet the system requirements. Here, we do not give a specific calculation formula for storage requirements, because the requirements of each sharding-based blockchain system are different.

### C. TRAINING ELM

After the feature selection, ELM is selected as the classifier to learn the five features: the objective feature, the objective feature associated with the node, the historical popularity, the hidden popularity and the storage requirements. The blockchain system detects and records the 4 knowledge (number of transactions, transaction contents, historical query records and security analysis) of each block, and calculates the feature value to form a feature array. The job of detection and record is finished by the verification nodes in Elasticchain [7]. Then, the arrays of features are used as inputs to train the ELM model. In the ELM-based classifier, each block can be classified into "hot" class or "non-hot" class.

### D. THE OPTIMIZED DATA STORAGE MODEL

We take Elasticchain as an example to illustrate the building process of the optimized data storage model, as shown in Algorithm 2.

```
Algorithm 2 The Optimized Data Storage Model
Input:   parameters $a$, $b$ and $T_f$
Output:  blockchain data storage scheme
1  verification nodes ($V$) visit blocks $\{b_1, b_2, \ldots, b_i\}$ at
   every same period of time;
2  $V$ record the data ($N_t$, $N_u$, $T_v$, $N_n$, $TV_n$, $TI$, $QT$, $DT$,
   and security requirement};
3  $V$ calculate the feature values
   ($OF$, $OFN$, $HIS$, $HID$, $SR$) of each block according to
   the recorded data and parameters;
4  $V$ train the ELM classifier by using feature values;
5  $V$ classify the new block (hot block or non-hot block);
6  $V$ record the classification results;
7  $V$ provide hot blocks for user nodes;
8  user nodes store hot blocks locally;
```

Firstly, The values of parameters $a, b$ and $T_f$ will be determined according to system requirements. Secondly, the verification nodes of Elasticchain system visit blocks $\{b_1, b_2, \ldots, b_i\}$ at every same period time and record their knowledge (each $N_t$, $N_u$ and $T_v$; each $N_n$ and $TV_n$; each $TI$ and $QT$; each $DT$; each security requirement). Then, the optimized data storage model calculates the five feature values of each block based on their knowledge. The five features are the objective feature of a block ($OF$), the objective feature of the block associated with the node ($OFN$), the historical popularity ($HIS$), the hidden popularity ($HID$) and the storage requirements ($SR$).

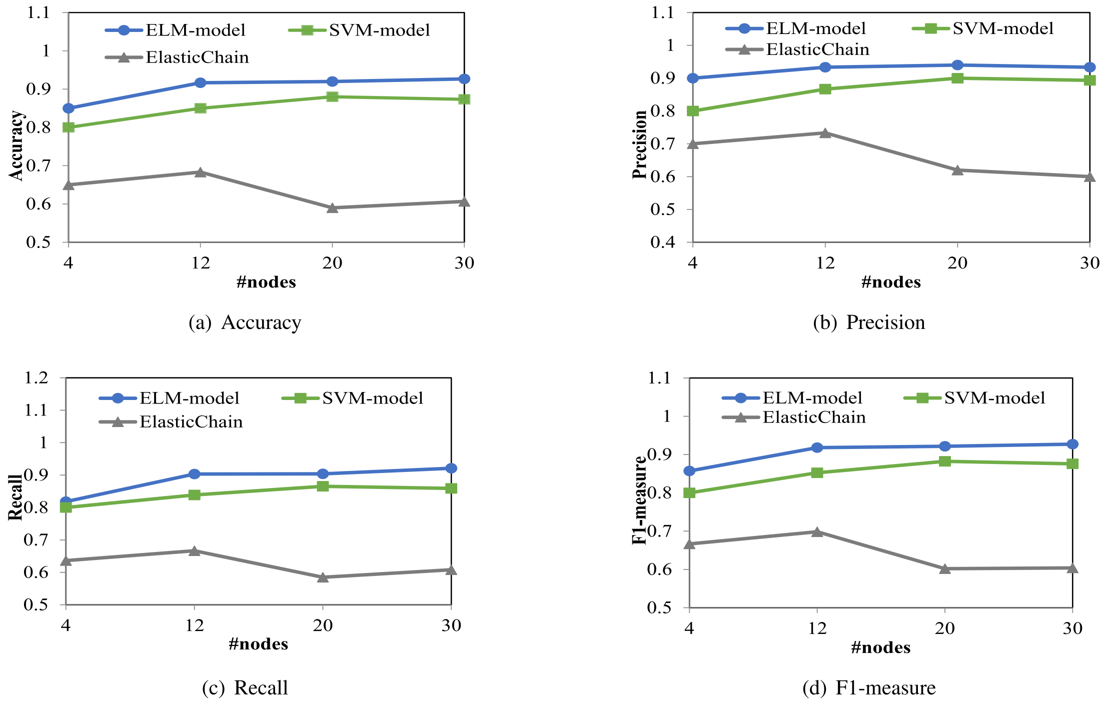
Next, ELM classifier trains the model by using these feature values. The trained model is used to classify new blocks as they are generated. Verification nodes will record the classification results (hot block or non-hot block). Finally, user nodes store the hot blocks locally for quick retrieval.

## V. EVALUATION

The setup of evaluations is firstly introduced in Section 5.1. Then we evaluate the classification performance of the optimized data storage model in Section 5.2. Section 5.3 evaluates the query performance of blockchain system by using the optimized data storage model.

### A. EXPERIMENT SETTINGS

All experiments are conducted on a 3.2-GHz, Core i5 CPU PC with 16G memory running the Window 7 operating system. Each node in blockchain system is created by VMware



(a) Accuracy

(b) Precision

(c) Recall

(d) F1-measure

**FIGURE 4. Experimental results on Dataset 2.**

Workstation. Each node is based on ubuntu16.04 system and is configured with 300MB of memory and 1GB of hard disk space.

The synthetic data is used as the experimental data set in this paper. We refer to the characteristics of the real blockchain system and use the controlled variable method to assign the parameters of 50 blocks. The parameters includes $N_t$, $N_u$, $T_v$, $N_n$, $TV_n$, $TI$, $QT$ and $DT$. In addition, $a = 1$ and $b = 1$. Then, 30 nodes are used in the experiment. Each block will generate a group of parameters for a node. For each group of parameters, four feature values ($OF$, $OFN$, $HIS$ and $HID$) will be calculated by formula (10), (11), (12) and (13), respectively. We ended up with 1,500 sets of data. Each group of data is artificially divided into hot blocks or non-hot blocks. The storage requirements feature are not considered in the experiment, because each sharding-based blockchain system has different evaluation criteria for the storage requirements of the blocks.

The real dataset is not used in the experiment because there is no real dataset that meets the experimental requirements. For example, supply chain and cold chain food traceability platforms (e.g. Beijing cold chain food traceability platform [18], etc.) are suitable application scenarios for our model, but most platforms have been established and applied recently, and data cannot be obtained. Or in some public blockchain systems (e.g. Bitcoin, Ethereum, and etc.), we can get the objective feature of each block, but other features such as OFN, HIS, and HID are not available. Therefore, synthetic data is used.

### B. EVALUATION OF CLASSIFICATION PERFORMANCE

In the experiment, 1500 groups of features of 50 blockchains will be divided into 3 groups. Dataset 1 contains the feature values of 40 blocks (1200 groups of features). Dataset 2 and Dataset 3 contain the feature values of 5 blocks (150 groups of features). 50% of the data in Dataset 2 is hot block for nodes, and 70% of the data in Dataset 3 is hot block. Dataset 1 is used as the training set and input to the optimization model based on ELM. Dataset 2 and Dataset 3 will be used as the test set of the model. The number of hidden layer nodes is 10 in the ELM classifier.

Then, the SVM method is compared with the ELM method in the experiment. We modify the ELM classifier in the optimization model to an SVM classifier for training and testing. We choose a sigmoidal kernel function and set the penalty parameter as 10 for the SVM-based classifier.

We experimented on the accuracy, precision, recall and F1-measure of block popularity evaluation. The accuracy of a classifier can be expressed as follows:

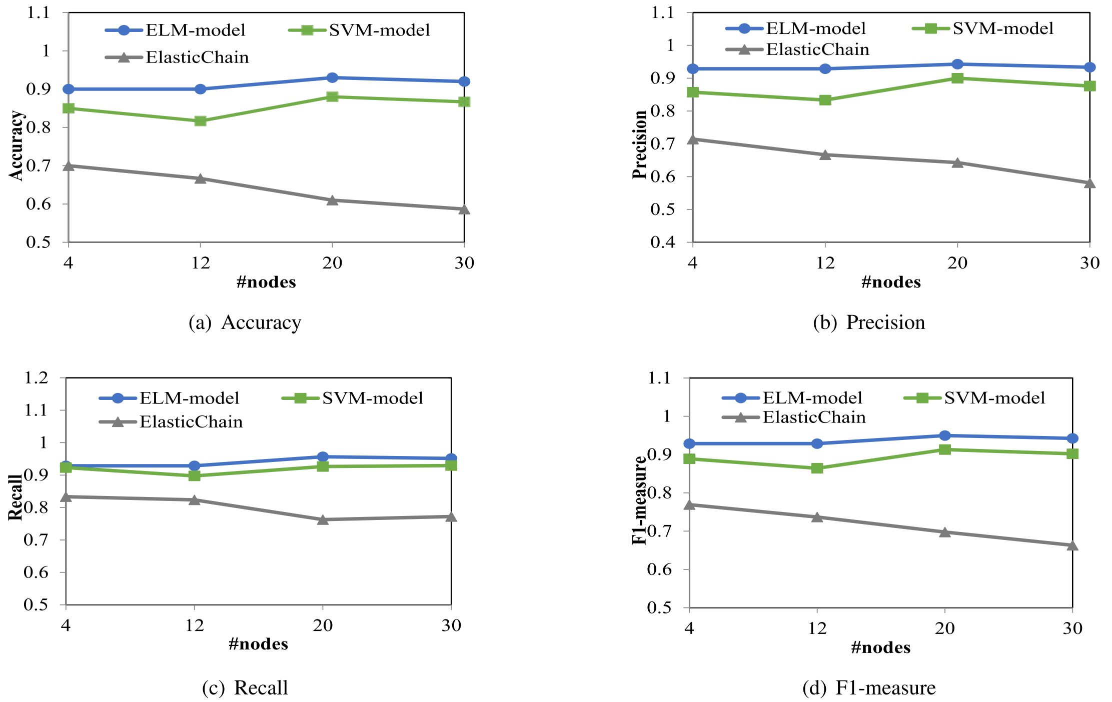
$$Accuracy = (TP + TN)/(TP + FN + FP + TN) \quad (14)$$

where $TP$ is True Positive, $FP$ is False Positive, $TN$ is True Negative, $FN$ is False Negative. And the precision of a classifier can be expressed as follows:

$$Precision = TP/(TP + FP) \quad (15)$$

The recall of a classifier can be expressed as follows:

$$Recall = TP/(TP + FN) \quad (16)$$



(a) Accuracy

(b) Precision

(c) Recall

(d) F1-measure

**FIGURE 5.** Experimental results on Dataset 3.

The F1-measure of a classifier can be expressed as follows:

$$F1 - measure = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \tag{17}$$

The optimized data storage model (ELM-model), SVM-based optimized model (SVM-model) and ElasticChain model are tested based on Dataset2 and Dataset3 when there are 4, 12, 20 and 30 nodes in models. Experimental results on Dataset 2 is shown in Figure 4 and Figure 5 shows the results based on Dataset 3.

We can get the following conclusions from Figure 4 and Figure 5.

(1) The four evaluation indexes (accuracy, precision, recall and F1-measure) of the optimized models (ELM-based and SVM-based) are obviously higher than that of ElasticChain model. The reason is that the optimized data storage model is adopted in ELM-based and SVM-based models to classify the hot blocks. The optimized models give a more comprehensive evaluation of the popularity of a block. Meanwhile, the evaluation indexes of ELM-based optimized data storage model are all the highest, and the indexes of SVM-based model is slightly lower than that of ELM-based model. This is because the performance of the ELM classifier is slightly better than that of the SVM.

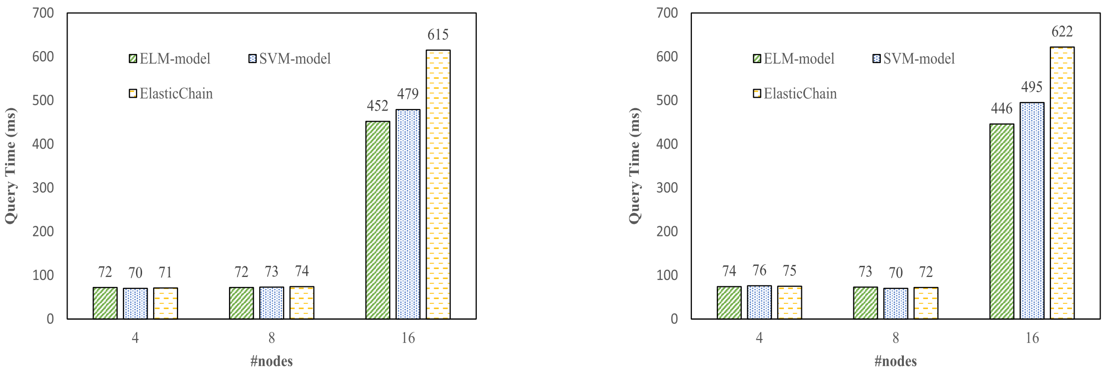
(2) In the same dataset, the accuracy, precision, recall and F1-measure of the block popularity evaluation show shows an upward trend as the number of nodes increases. This is due to the fact that there is less feature data for blocks when the number of nodes is small. As the number of nodes increases, the data of block popularity increases continuously. When the amount of test data is large, the classification results are more convincing.

(3) By comparing Dataset 2 and Dataset 3, we can find that the accuracy, precision, recall and F1-measure of the block popularity evaluation based on Dataset 3 are slightly higher than those based on Dataset 2. However, the difference is so small that it is negligible. Therefore, different datasets have little effect on the performance of the optimization model.

### C. EVALUATION OF QUERY PERFORMANCE

Then, we test the query performance of sharding-based blockchain systems when using the optimized data storage model. We build the ElasticChain system by using Hyperledger fabric V0.6 because fabric V0.6 is one of the earliest widely used blockchain systems. Three ElasticChain systems are deployed in the experiment. The first system adopts the ELM-based optimized data storage model. The second system adopts the SVM-based optimized model, and the third system is the ElasticChain original system. According to the benchmark work of the blockbench [19], when the system is running normally, the maximum number of nodes is 16. Thus, 4, 8, 12 and 16 nodes are established.

We operate the chaincode called *example*02.*go* [20], and every time a transaction is completed, 5.39KB broadcast message is generated. The block size is set to 100. In other words, each block contains 100 blocks completed. When 2000 transactions are completed, 20 blocks will be created and stored by nodes according to the duplicate ratio regulation algorithm [7]. We randomly select 10 transactions for query and record the query time.



(a) Dataset 2

(b) Dataset 3

**FIGURE 6. The average query time for each transaction.**

The average time for querying a transaction based on Dataset 2 and Dataset 3 is shown in Figure 6, and we can draw the conclusions as below.

(1) The average query time for a transaction when there are 16 nodes in three systems (ELM-based optimized model, SVM-based optimized model and ElasticChain) is much larger the query time when there are 4 or 8 nodes. The reason is that we set the minimum number of block duplicates as 8 in the duplicate ratio regulation algorithm. When fewer than 8 nodes exist in the system, each node maintains a complete blockchain copy, and the queries in three models are similar to local queries. However, when there are more than eight nodes in the system, sharding blockchain data is stored in nodes. Nodes may need to visit other nodes to retrieval the target data and the response times are increased significantly.

(2) When 4 and 8 nodes exist in three systems, the average query time of the three systems is almost the same because they all use the local query method. However, when there are 16 nodes in systems, the average query time of ELM-based and SVM-based optimized model is lower than that of ElasticChain system. The reason is optimized models adopt the feature extraction method based on this paper, and the blocks with high popularity are saved in the nodes. Optimized models reduce the number of cross-node queries and increase the number of local queries. Therefore, the query time is reduced. Meanwhile, the query response time of the ELM-based optimized model is less than the time of SVM-based optimized model because of the high performance of the ELM model.

## VI. CONCLUSION

In our study, we presented the optimized data storage method for sharding-based blockchain. The optimized method combines blockchain and artificial intelligence and solves the current hot problem that the current cold chain information is difficult and inefficient to trace. Five features are proposed in this paper to evaluate the popularity of a block, including the objective feature of a block, the objective feature of the block associated with the node, the historical popularity, the hidden popularity and the storage requirements. Then the ELM method is used to be the classifier. The experimental results on synthetic data demonstrate the accuracy and efficiency of the optimized data storage model.

In the future, more sharding-based blockchain systems will be designed. We need to analyze the different characteristics of each new system and propose suitable data storage methods for each system. At the same time, machine learning techniques are researched by many experts and large companies. Novel and efficient machine learning methods are constantly being proposed. In the future, it will be an attractive direction to replace the ELM method with other more effective machine learning methods in sharding-based blockchain systems.

## REFERENCES

[1] B. H. Yang and C. Chen, *Blockchain Principle, Design and Application*, 1st ed. Beijing, China: China Machine Press, 2020.

[2] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy*, San Francisco, CA, USA, May 2018, pp. 583–598.

[3] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 2018, pp. 931–948.

[4] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Vienna, Austria, Oct. 2016, pp. 17–30.

[5] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proc. SIGMOD*, Amsterdam, The Netherlands, Jun. 2019, pp. 123–140.

[6] Z. Xu, S. Han, and L. Chen, "CUB, a consensus unit-based storage scheme for blockchain system," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Paris, France, Apr. 2018, pp. 173–184.

[7] D. Jia, J. Xin, Z. Wang, W. Guo, and G. Wang, "ElasticChain: Support very large blockchain by reducing data redundancy," in *Proc. APWeb-WAIM*, Macau, China, 2018, pp. 440–454.

[8] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Honolulu, HI, USA, Jun. 2017, pp. 468–475.

[9] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.

[10] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.

[11] C. Li, C. Deng, S. Zhou, B. Zhao, and G.-B. Huang, "Conditional random mapping for effective ELM feature representation," *Cogn. Comput.*, vol. 10, no. 5, pp. 827–847, Oct. 2018.

[12] D. Cui, G.-B. Huang, and T. Liu, "ELM based smile detection using distance vector," *Pattern Recognit.*, vol. 79, pp. 356–369, Jul. 2018.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[14] G.-B. Huang and C. K. Siew, "Extreme learning machine: RBF network case," in *Proc. Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, Kunming, China, 2004, pp. 1029–1036.

[15] *Extreme Learning Machine*. Accessed: Jan. 23, 2021. [Online]. Available: https://personal.ntu.edu.sg/egbhuang/

[16] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[17] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.

[18] *Beijing Cold Chain Food Traceability Platform*. Accessed: May 5, 2021. [Online]. Available: https://sp.scjgj.beijing.gov.cn/cctp/login

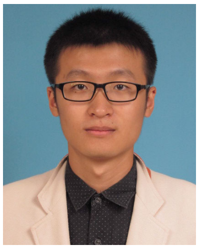
[19] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "BLOCKBENCH: Framework for analyzing private blockchains," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, Chicago, IL, USA, 2017, pp. 1085–1100.

[20] *Beijing Cold Chain Food Traceability Platform*. Accessed: May 5, 2021. [Online]. Available: https://github.com/hyperledger/fabric/tree/v0.6

[21] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proc. 2nd Int. Conf. Internet-of-Things Design Implement. (IoTDI)*, Pittsburgh, PA, USA, 2017, pp. 173–178.

[22] A. Dorri, S. S. Kanhere, and R. Jurdak, "MOF-BC: A memory optimized and flexible blockchain for large scale networks," *Future Gener. Comput. Syst.*, vol. 92, pp. 357–373, Mar. 2019.

[23] V. Dedeoglu, R. Jurdak, A. Dorri, R. Lunardi, R. Michelin, A. F. Zorzo, and S. Kanhere, "Blockchain technologies for IoT," in *Advanced Applications of Blockchain Technology*. Singapore: Springer, 2020, pp. 55–89.



**DAYU JIA** received the B.Sc. and M.Sc. degrees in computer science and technology from Northeastern University, Shenyang, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree in computer science and engineering. He studied at the National University of Singapore as a joint training of doctoral student, from October 2018 to April 2020. His research interests include scalable storage for blockchain and big data.

**JUNCHANG XIN** received the B.Sc., M.Sc., and Ph.D. degrees in computer science and technology from Northeastern University, China, in 2002, 2005, and 2008, respectively. He visited the National University of Singapore as a Postdoctoral Visitor, from 2010 to 2011. He is currently an Associate Professor with the School of Computer Science and Engineering, Northeastern University. He has published more than 60 research articles. His research interests include big data, uncertain data, bioinformatics, and blockchain database. He served as PIs or Co-PIs for more than ten national research grants from NSFC, the 863 Program, the Project 908 under the State Oceanic Administration, and so on.

**ZHIQIONG WANG** received the M.Sc. degree in computer applications technology and the Ph.D. degree in computer software and theory from Northeastern University, China, in 2008 and 2014, respectively. She visited the National University of Singapore, in 2010, and The Chinese University of Hong Kong, in 2013, as an Academic Visitor. She is currently an Associate Professor with the College of Medicine and Biological Information Engineering, Northeastern University. She has published more than 30 articles. Her current research interests include biomedical, biological data processing, cloud computing, and machine learning. She served as PIs or Co-PIs for more than ten national research grants from NSFC, the Natural Science Foundation of Liaoning Province, and so on.

**GUOREN WANG** received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Computer Science, Northeastern University, China, in 1988, 1991, and 1996, respectively. He is currently a Professor with the Department of Computer Science, Beijing Institute of Technology, China. He has published more than 100 research articles. His research interests include XML data management, query processing and optimization, bioinformatics, high-dimensional indexing, parallel database systems, and P2P data management.