S-MBDA: A Blockchain-based Architecture for Secure Storage and Sharing of Material Big-Data

Ran Wang, *Graduate Student Member, IEEE,* Cheng Xu, *Member, IEEE,* Fangwen Ye, Sisui Tang, and Xiaotong Zhang, *Senior Member, IEEE*

***Abstract*—Material data forms the foundation of the Industrial Internet of Things (IIoT). The rapid advancement of big data technology has opened up new opportunities for material research and development, ushering in the era of data-driven paradigms. As the cornerstone for material genetic engineering technology, the material big data platform is expanding its data scale and facing an increasing demand for sharing in light of the continuous progress and widespread application of big data technology. However, this development also poses security challenges, including the risks of data leakage and tampering. To address these challenges, this article focuses on the National Materials Genetic Engineering Discrete Data Exchange Platform (MGED). It leverages blockchain technology to design a secure material big-data storage and sharing architecture, S-MBDA, ensuring the security and reliability of the material's big data platform. Additionally, a verifiable retrieval scheme based on a two-layer index structure of bitmap and MPT tree is proposed to enhance the efficiency of blockchain-based retrieval. This scheme aims to guarantee the integrity of retrieval data while achieving efficient and accurate searches across heterogeneous data sources. Through integrating blockchain technology and adopting a novel retrieval scheme, the article presents a comprehensive approach to secure material data storage, sharing, and retrieval. The proposed architecture and scheme address the critical security concerns associated with material big data platforms and contribute to the efficient and accurate retrieval of heterogeneous data.**

***Index Terms*—materials big-data; data storage; data sharing; blockchain; secure multiparty computation.**

I. Introduction

The Industrial Internet of Things (IIoT) enables the interconnection of physical and digital worlds, resulting in the deployment of cyber-physical systems (CPS) within manufacturing industries [1], [2]. Material data serves as the foundation for the IIoT, and the rapid development of big data technology has brought new opportunities for material research and development, gradually shifting towards a data-driven paradigm [3]. Leveraging data-driven techniques can significantly reduce the R&D cycle and costs simultaneously.

This work is supported in part by the National Key Research and Development Program of China under Grant 2021YFB3702403, in part by the National Natural Science Foundation of China under Grant 62101029, and in part by the China Scholarship Council Award under Grant 202006465043. (*Corresponding authors:* Cheng Xu and Xiaotong Zhang)

The authors are with School of Computer and Communication Engineering, University of Science and Technology Beijing. Ran Wang, Cheng Xu, and Xiaotong Zhang are also with Beijing Advanced Innovation Center for Materials Engineering, Shunde Innovation School, and State Key Laboratory for Advanced Metals and Materials, University of Science and Technology Beijing (email: wangran423@foxmail.com; xucheng@ustb.edu.cn; yfwen2000@outlook.com; tangsisui@163.com; zxt@ies.ustb.edu.cn).

The emergence of Materials Genome Engineering (MGE) [4] marks a pivotal moment in data-driven materials science. MGE combines artificial intelligence, machine learning, high-throughput computing, and other technologies to accelerate the material research and development process by predicting and designing material properties and structures [1]. Material genetic engineering relies on large volumes of material data to train models and validate algorithms. However, extracting actionable insights from the collected data necessitates an infrastructure that ensures data trustworthiness [5]. Consequently, many countries have initiated the construction of material data infrastructure [6]–[8], leading to the development of various material database platforms. These databases encompass diverse data types, including experimental data, calculation results, and literature information. The data is archived, classified, and stored in a unified database through standardized data grids and metadata, enabling convenient search, analysis, and sharing for researchers and engineering technicians [3]. These databases integrate a vast amount of material data, providing crucial support and scientific services for material genetic engineering research. However, with the increasing volume of data and the demand for data sharing, material database platforms face significant challenges regarding security and trustworthiness. In general, the following challenges arise:

1) The data is hosted on the central platform, and users have no autonomous control rights;
2) The data is susceptible to weak information security controls, making it vulnerable to local attacks or issues, potentially resulting in data leakage.
3) The operation records of the data lack comprehensive openness and transparency, leading to a weak ability to audit and trace.

The emergence of blockchain technology offers new solutions to these challenges. The fundamental concept of blockchain technology revolves around decentralization and tamper resistance, ensuring secure transactions while promoting data sharing and exchange. By integrating blockchain technology with MGED, the security, trusted storage, and sharing of material data can be guaranteed. Blockchain technology enables traceability and fairness in material data sources and usage records, thereby enhancing the credibility and security of the data. Additionally, decentralized storage and sharing of material data can be achieved, fostering data collaboration and cooperation in material genetic engineering.

This paper presents a secure sharing platform for mate-

rial big data based on blockchain, leveraging the MGED framework. Serving as a pivotal component of MGED's data applications, our platform grants users access to an extensive collection of material data resources from over thirty research institutions in China. The platform effectively addresses the significant challenges associated with storing and utilizing diverse data from various sources. It facilitates service sharing, expedites material discovery, and caters to the demands of high-throughput calculations and experiments. The key contributions of this paper can be summarized as follows:

1) We propose a blockchain-based secured management architecture for MGED, addressing the challenges in data storage management and security. The framework utilizes blockchain technology to establish a distributed, secure storage system for big data. With this approach, participants can deploy blockchain nodes flexibly without modifying the underlying database framework. This enables seamless joining or exiting of the platform for data providers and consumers while facilitating unified management of databases with different types. Moreover, the distributed ledger ensures data storage security by offering features such as data tamper-proofing, traceability, and auditability.
2) To overcome the efficiency limitations of blockchain-based retrieval, we propose a verifiable retrieval scheme that leverages a two-layer index structure comprising bitmaps and Merkle Patricia Trie (MPT). This scheme aims to ensure the integrity of retrieval data while achieving an efficient and accurate search for heterogeneous data. Experimental results demonstrate that the system exhibits high levels of security, reliability, and availability, significantly improving retrieval efficiency compared to traditional blockchain retrieval solutions.
3) The paper presents a blockchain-based retrieval and multiparty secure computing method that enables secure sharing and utilization of heterogeneous material data. Leveraging the tamper-proof nature of blockchain, this approach guarantees the reliability and availability of data retrieval. Additionally, it incorporates techniques such as horizontal federated learning, secret sharing, and homomorphic encryption to achieve "available but invisible" data. Therefore, it mitigates the risk of data leakage during the sharing process and fulfills the requirements of joint modeling and prediction of new materials.

The remainder of this paper is organized as follows. In Section II, we present an overview of the current state-of-the-art and challenges in material big-data sharing platforms. Section III showcases the architecture of our secured big-data sharing platform, Secure-MBDF, specifically designed for material genome engineering. The construction details and discussions are provided in Section IV. Finally, Section V offers a summary of the paper and discusses future prospects.

II. Related Work

With the rapid expansion of the big-data industry, the open sharing, exchange, and circulation of big data have gradually become a trend, promoting the release of data value in various fields. Sectors such as finance, electricity, transportation, and the industrial Internet of Things have already established big-data sharing platforms with solutions tailored to their data characteristics [9]–[11]. However, in the field of materials, the development of big-data platforms is not yet mature, and there is a lack of security mechanisms to ensure the safety of material data sharing. The multi-modality, isomerization, and discreteness characteristics of material data pose significant challenges that need to be addressed [12]. Different types of materials have distinct compositions and performance concerns. Even the same material often exists in various structural forms across different databases. The severe fragmentation, isomerization, and decentralization of material data make it exceedingly difficult to collect, store, and utilize effectively. Furthermore, the issue of security in material big-data sharing is of vital concern to academia and industry alike. Owners of material data often possess sensitive data that represent valuable assets, making it challenging to transfer and resulting in the formation of "data islands." This scarcity of high-quality material data hinders material science research in research institutions, ultimately affecting the efficient development of the material industry.

To expedite the process of material research and development, several studies have focused on the construction of material big-data platforms. For instance, AFLOW [6] is a large database based on high-flux first principles. It encompasses 12 applications, including AFLOW π, AFLOW-ML, PAOFLOW, and etc, which can screen the structure and properties of materials. COD [13] collects "small molecule/small to medium unit cell" crystal structures through an open-access distribution model, making them freely available on the Internet. The Materials Data Facility (MDF) [14], based on DSpace and Globus systems, offers two cloud hosting services, data publishing and data discovery. It enables open data sharing, self-service data publishing, and management and encourages data reuse. The Materials Project [15] hosts a database with a vast amount of information, including nearly 60,000 crystal structures, and allows users to search and filter materials through a database interface by writing code. Table I compares and analyzes the advantages of our material big-data security sharing platform compared to state-of-the-art material big-data platforms at home and abroad, considering aspects such as data collection, storage, utilization, and security.

Based on the summary analysis in Table I, it is evident that material big-data platforms still lack robust security mechanisms. These platforms typically provide basic security features such as authentication and access control. For instance, COD [18] ensures auditability of data operations through logging and data backup, while MARVEL NCCR [7] employs a directed acyclic graph to guarantee data traceability and system robustness. Although these mechanisms offer good protection for static data, they fall short of effectively addressing privacy concerns during data sharing, including issues related to traceability and auditability. During the sharing process, participants may attempt to infer others' private data from the shared data, potentially resulting in the leakage of sensitive information and compromising the rights and interests of data providers.

TABLE I: A summary of existing typical material big-data platforms.

Name	Description	Storage	Utilization	Security
AFLOW [6], [16], [17]	AFLOW has 12 applications including AFLOW π, AFLOW-ML and PAOFLOW, which can screen the structure and properties of materials	Centralized storage	Retrieval and prediction	—
Crystallography Open Database (COD) [13], [18]	COD collects all known "small molecule / small to medium unit cell" crystal structures.	Centralized storage	Retrieving and downloading	Logging, access control, data backup
MARVEL NCCR [7]	Material informatics platform for data-driven high-throughput quantum simulation. Supported by aiida infrastructure.	Centralized storage	Retrieval and simulation modeling	traceability and system robustness
The Materials Data Facility (MDF) [14]	Based on DSpace and Globus systems, MDF operates two cloud hosting services, data publishing, and data discovery.	Distributed storage	Retrieval, data aggregation, and automated analysis.	Identity authentication, access control, disaster recovery backup
Materials Project [15]	The Materials Project contains a database with a large amount of information.	Distributed storage	retrieval, download, analysis, and design	Identity authentication and data integrity verification
NOMAD CoE [8]	Provide complete input and output file storage of all important computational material science codes, and build multiple big-data services at the top.	Centralized storage	Retrieve and download	Identity authentication and access control
Open Quantum Materials Database (OQMD) [19]	OQMD is a database based on density functional theory (DFT) to calculate material thermodynamics and structure.	Based on ICSD database	Retrieval and data analysis	—
Open materials database [20]	The open materials database uses a high-throughput toolkit to provide a free open source framework for calculating and analyzing results.	Based on COD database	calculation and analysis	—
AtSteel [21]	Provide standard data and experimental data of steel, welding materials, and non-ferrous metals.	Centralized storage	Retrieval	Identity Authentication
Our proposed architecture	A material science data system and a material science data sharing service platform that meet different national needs.	Adopt "transaction info stored on-chain, and original data stored off-chain."	Retrieval and download, digital identification, secure multiparty computing.	Identity authentication, access control, tamper-proof, security audit, and traceability.

The emergence of blockchain technology has opened up new possibilities for addressing these challenges. The academic community has been exploring the use of trusted blockchain networks to record data attribution and access permissions, thereby ensuring the lawful utilization of data [22]. For instance, Chen et al. [11] proposed a blockchain-based secured big-data sharing model that synchronizes blockchain information among various nodes, ensuring the auditability and traceability of data sharing. However, due to the inherent openness and transparency of blockchains, additional technologies have been introduced to enhance data-sharing security. Yang et al. [23] presented a data tamper-proof mechanism based on blockchain, incorporating cryptographic algorithms to prevent tampering of transaction data during user storage, thereby ensuring transaction security and data reliability. Similarly, Sex. et al. [24] utilized blockchain-based secure multiparty computation to achieve privacy-protected data sharing. These methods offer certain levels of security and privacy for the original data. However, they still face challenges related to result leakage and tampering during function invocation.

Additionally, direct data sharing poses risks such as the data owner losing control over the data and the potential for dishonest participants to share it with unauthorized entities. The integration of federated learning into the blockchain consensus process enables the sharing of data models while maintaining control over the raw data [25]. However, a critical challenge that remains to be urgently addressed is the potential leakage of parameters in the federated learning model, which could allow inference of the original data. To enhance the security of federated learning, the academic community has explored the use of secure multiparty computing. Wei et al. [26] proposed a framework based on Differential Privacy (DP), which adds artificial noise to the training parameters before federated model aggregation, thereby protecting the security of the model parameters. Chai et al. [27] introduced a knowledge-sharing federated learning algorithm based on a hierarchical blockchain. The layered blockchain framework enhances the reliability and security of knowledge sharing. However, it exhibits weaknesses when dealing with storage problems that require significant resource consumption.

Taking all of these factors into consideration, we have developed a secured big-data sharing platform for materials genome engineering. This platform aims to enhance data utilization, promote material data sharing, expedite the material discovery, and cater to the data requirements of high-throughput computing and experiments for designing new materials.

III. Secured Big-Data Sharing Platform for Materials Genome Engineering

In this section, we will discuss the overall architecture of our platform, the data storage framework based on blockchain, an efficient and verifiable retrieval algorithm, and multiparty collaborative services. The architecture we propose enables data retrieval, and multiparty collaborative calculations, and

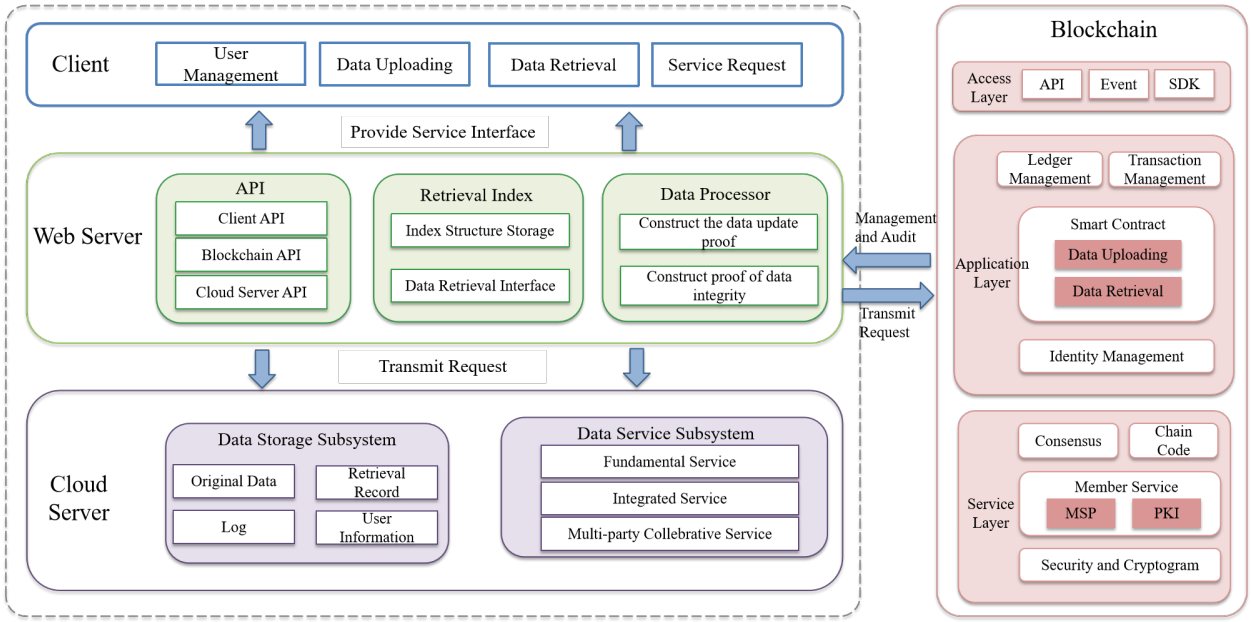


Fig. 1: The architecture diagram of our proposed secured material big-data sharing platform.

TABLE II: Notation.

Symbol	Description
$Data_{update}$	The object obtained by combining metadata and data content hash
DataArr	Array, whose members are the $Data_{update}$ of the data uploaded by the user
DataHsh	Constructed by a web server, taking hash values for DataArr
RootHash	MPT root hash corresponding to material classification
VO_{update}	Proof of web server construction when uploading data, provided to blockchain
cID	Material classification ID
W	Keyword set
$cIdList$	Material classification ID set
$R_{(chain,cId)}$	The RootHash of MPT corresponding to the material categories stored on the blockchain
VO_{cId}	Proof constructed by the web server for the corresponding material category ID during retrieval, provided to the client
$Map < cId, R_{(chain,cId)} >$	Hash table, where key is the material category id and value is the RootHash of the corresponding MPT
$Map < cId, R_{(chain,cId)}, VO_{cId} >$	Hash table, where key is the material category id and value is the set of root hashes and proofs corresponding to the MPT

meets the application requirements for material data prediction, modeling, and discovery. Symbol descriptions relevant to this section are provided in Table II.

A. The Overall Architecture

Based on the original underlying architecture, this paper constructs a secured big-data sharing platform for material genome engineering by integrating the Hyperledger Fabric consortium chain [28]. The platform offers an open and collaborative environment for researchers to conveniently and securely share, retrieve, calculate, and analyze data. The framework diagram of the secured big-data sharing platform is illustrated in Fig. 1, consisting of the client, blockchain, cloud server, and web server, which provide central services connecting these components.

Client: The client serves as the entry point for users to interact with the system and provides services through web pages. It encompasses functionalities such as data upload, data retrieval, and other service requests in the data service subsystem of the cloud server. Additionally, it includes user management functions, which comprise user authentication interfaces and user login pages.

Web Server: The web server acts as the middleware within the platform architecture. It receives upload, retrieval, or other transaction requests from client-side APIs and then assigns transaction tasks to blockchain and cloud server through their respective APIs. The advantage of our proposed blockchain framework is that users are not required to comprehend the underlying architecture of the platform, thereby reducing cognitive load and learning costs. Moreover, our architecture provides a more general solution, enhances the scalability of the blockchain, and can serve as a reference for big data-sharing platforms in other industries or domains.

Blockchain: The blockchain framework, illustrated in Fig. 2, establishes a decentralized distributed network architecture for diverse organizations. Within this system, the blockchain is implemented through Hyperledger Fabric, offering distinct advantages over public blockchains like Ethereum. These advantages include robust permission control, support for consortium chains, smart contracts, customizability, and modularity, rendering it an ideal choice for a material big data platform involving multiple research institutions. Hyperledger Fabric enables programming with smart contracts, facilitating the execution of customized business logic directly on the blockchain. This capability proves crucial for managing intricate transactions and contracts within the material big data platform. Participants within the blockchain have the ability to submit smart contracts, enabling them to perform a range of tasks such as identity verification, data upload, data retrieval, and model aggregation, thereby fulfilling diverse

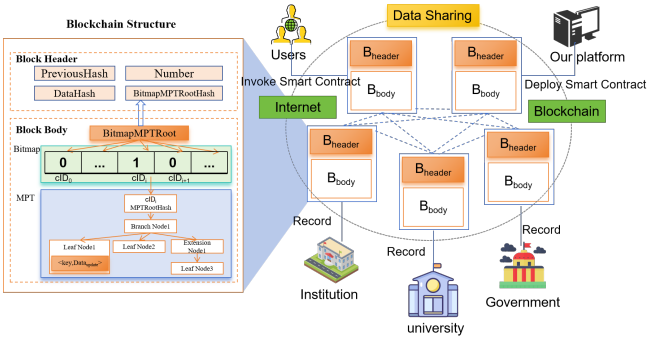


Fig. 2: The architecture diagram of our proposed secured material big-data sharing platform.

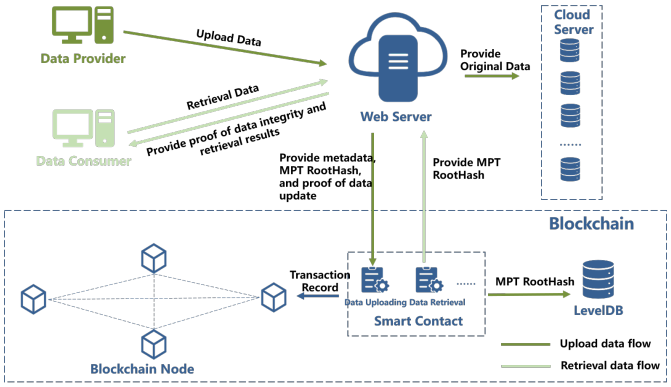


Fig. 3: The secured big-data storage framework based on blockchain.

functions. Smart contracts are validated by all participants on the blockchain, and once a consensus is reached, the smart contracts are deployed on the blockchain. Smart contracts are the sole means of creating transactions on the blockchain. The material big data platform and other participants on the Fabric consortium chain can call the relevant smart contracts through locally deployed blockchain nodes to perform permission authentication, data storage, and retrieval on the blockchain. Each organization deploys at least one blockchain node. The blockchain primarily stores metadata, data content hashes, and transaction records. To achieve efficient and verifiable retrieval based on the blockchain, we modified the blockchain's block structure, as shown in Fig. 2.

From the perspective of individual block structure, each block comprises a block header (Bheader) and a block body (Bbody). The block body (Bbody) houses a two-layer index structure, combining Bitmap and Merkle Patricia Trie (MPT). The first layer, the Bitmap index, facilitates retrieval based on material types, while the second layer constructs the corresponding MPT structure for a specific material type. Within the MPT, leaf nodes store data in a "key-value" format, where the value (Dataupdate) encompasses not only metadata for the original data but also the hash value of the data content. This specific index structure is stored in LevelDB, integrated into the blockchain for efficient retrieval. In contrast to the blockchain's original block header, we introduced a new field in the Bheader structure called BitmapMPTRootHash, utilized for subsequent data retrieval integrity verification. By implementing a two-layer index structure within the blockchain, the time complexity for keyword-based retrieval was reduced from $O(n^2)$ to $O(\log(n))$, enhancing retrieval efficiency while reducing the time needed for deserializing transaction records to extract metadata. Furthermore, the bitmap-based index structure supports range-based searches based on material types. This added index structure is highly applicable to both off-chain storage systems and on-chain material data retrieval.

Cloud Server: The cloud server provider primarily encompasses the data storage and service subsystems, which deliver data life-cycle services. The data storage subsystem stores the original data, parsed by the dynamic container module [29], into different data structures and provides formatted data to the client and the data service subsystem. The data storage subsystem is implemented using MongoDB, which supports various types of data storage, including metadata, management data, text data, structured data, binary data, and more. MongoDB offers robust scalability and supports multiple data type indexes, making it well-suited for storing heterogeneous materials data from various sources. Additionally, MongoDB supports the storage of all data types dynamically parsed from containers. The data service subsystem offers essential functionalities such as data retrieval, querying, multiparty collaborative computing, third-party integration, and other services for users. A bidirectional data flow between the data service and storage subsystems forms a virtuous circle of data sharing and service provision.

B. Data Storage Framework based on Blockchain

To ensure secure management throughout the data life cycle, our proposed architecture employs the Hyperledger Fabric consortium chain to handle the heterogeneous material data. This approach enables comprehensive data auditing and guarantees data integrity and availability.

The storage architecture for our platform is illustrated in Fig. 3. Since most material data is sensitive and voluminous, storing all the data on a blockchain with limited space would be resource-intensive and risky. Therefore, we utilize the blockchain for data management and retrieval, considering privacy concerns and storage limitations. Specifically, we adopt an approach where "transactions are stored on-chain, while original data is stored off-chain." This approach offers high security and throughput. Upload records and query records are stored on the blockchain, ensuring transparency and security throughout the process and enabling data tamper-proofing, traceability, and auditability. The original data is stored locally or uploaded to the cloud server by its owner, alleviating the computational burden on the blockchain. Fig. 3 depicts two primary material data flows: data uploading and data retrieval. These flows form a closed loop. The following sections illustrate the overall working mechanism of our blockchain-based secured big-data storage framework, using data upload and query as examples.

*1) **Data uploading**:* Data upload begins with user permission verification. If a user has the upload permission, they can

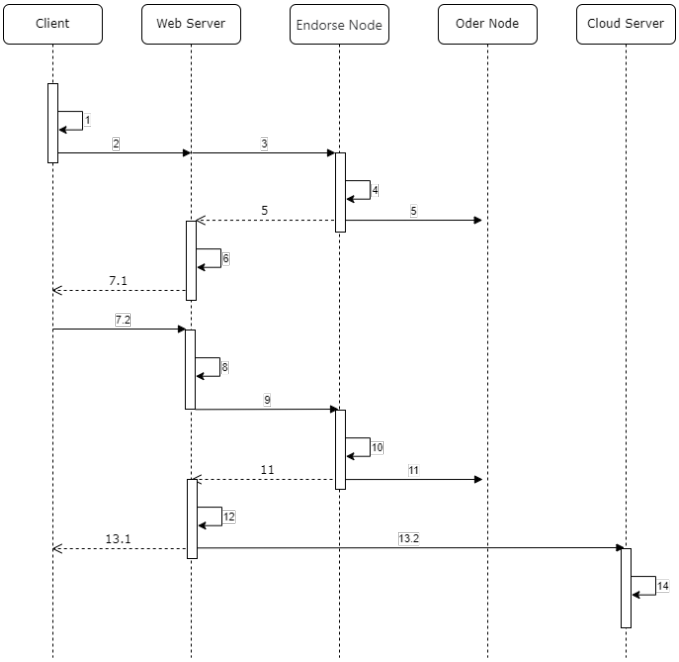


Fig. 4: The timing chart of data uploading.

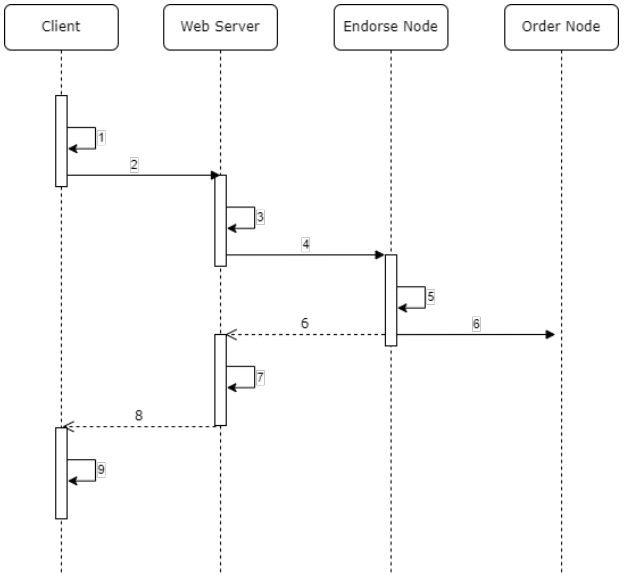


Fig. 5: The timing chart of data retrieval.

initiate the data upload process, as depicted in Fig. 4. The detailed steps for each stage are as follows:

1. Users fill out data forms on the client and click submit.
2. The client sends an authentication request with empty content, and the request header includes user information such as the username and user identifier.
3. The web server receives the request and forwards it to the blockchain.
4. Upon receiving the user's identity information in the blockchain network, the endorsement nodes determine whether the user has the required permission.
5. The endorsement node returns the endorsement results to the web server, while the sorting nodes in the blockchain network package the transactions, generate blocks, and broadcast them in the blockchain network to save the newly generated blocks.
6. The web server determines the authentication success based on the endorsement results.

7.1. If authentication fails, the process ends, and a corresponding prompt is displayed on the client.

7.2. If authentication is successful, the client sends an upload data request to the web server.

8. The web server processes the data by first calculating the hash of the data content, synthesizing it with the data metadata part to obtain $Data_{update}$, and updating the retrieval index structure. It also calculates VO_{update}.
9. VO_{update}, $Data_{update}$, and $R_{(chain,cId)}$ are transferred together to the blockchain network. The retrieval index structure and specific fields of VO_{update} will be described in "Subsection C."
10. Upon receiving the transaction, the endorsement node in the blockchain network executes a verification chain code to verify the legality of VO_{update}. If the verification is successful, the $R_{(chain,cId)}$ RootHash of the MPT corresponding to the material category to which the data belongs in LevelDB is updated. This step aims to achieve verifiable retrieval. The specific verification process will be explained in "Subsection C."
11. The endorsement node returns the execution result of the chain code to the web server, while the sorting node packages the verification result into a block and broadcasts it in the blockchain network.
12. The web server receives the data returned by the blockchain and determines the success of the transaction upload.

13.1. If the transaction fails, the process terminates, and a corresponding prompt is displayed on the client.

13.2. If the transaction is successful, an upload data request is sent to the cloud server.

14. The cloud server saves the full text of the data to the database.

*2) **Data retrieval**:* Data retrieval primarily focuses on keyword retrieval of the original data. However, the retrieval results only return metadata fields to safeguard data privacy. The process is illustrated in Fig. 5, and the detailed steps are as follows:

1. Users input search keywords on the client search page and click submit.
2. The client sends a retrieval request.
3. The web server searches the data index structure based on the keywords to obtain $cIdList$ and the set of metadata.
4. The web server forwards $cIdList$ to the blockchain network.
5. Upon receiving the transaction creation request, the endorsement node in the blockchain network first executes a data retrieval chain code. This code retrieves the corresponding $R_{(chain,cId)}$ based on each cId in $cIdList$ and adds the result to the $Map < cId, R_{(chain,cId)} >$ list.

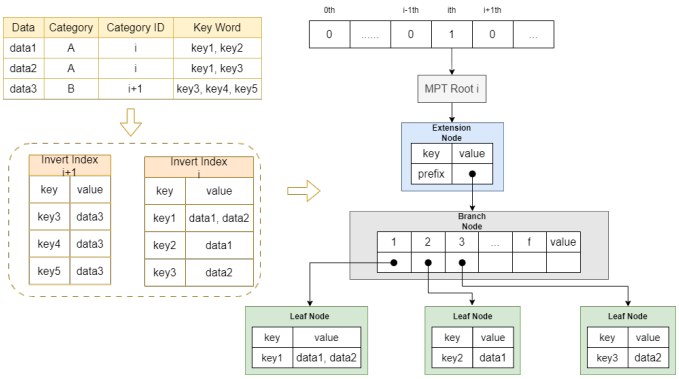


Fig. 6: An sample image of retrieval index structure.

6. The execution result of the chain code is packaged into a block, returned to the web server through the endorsement node, and broadcasted in the blockchain network.
7. The web server receives the retrieval transaction results returned by the blockchain, along with $Map < cId, R_{(chain,cId)} >$. It constructs VO_{cId} for each cId and adds the results to $Map < cId, R_{(chain,cId)}, VO_{cId} >$.
8. The web server returns $Map < cId, R_{(chain,cId)}, VO_{cId} >$ along with the metadata collection to the client.
9. The client performs retrieval verification and displays the search and verification results on the page.

C. An efficient and verifiable retrieval algorithm

In this subsection, we will present the retrieval index structure and retrieval integrity verification, building upon the data retrieval process described in subsection B.

1) Retrieval Index Structure: Our proposed material big data security sharing platform categorizes material science data into different material classifications. When uploading data, submitters must manually specify the material classification of the data, ensuring that all data on the platform has a unique material classification. Leveraging this characteristic, the retrieval index consists of two layers: the first is the material classification bitmap, and the second is the metadata Merkle Patricia Tree (MPT) corresponding to each material classification. Please refer to Fig. 6 for visualization.

A bitmap is an array of bits where each bit is assigned a corresponding value. The metadata Merkle Patricia Tree (MPT) represents a set of metadata values structured as an MPT tree. Our system creates a separate MPT tree for each material classification. The maintenance of the corresponding MPT tree is based on the inverted index. The standard leaf nodes in the MPT tree are presented as a list of [key, value], where the key is a unique hexadecimal code, and the value is an RLP code. By utilizing the inverted index of keyword data, the key (keyword) and value (metadata) are encoded and subsequently added to the corresponding MPT tree.

Currently, our proposed platform has a total of 138 material classifications. To ensure the scalability of the index structure and minimize storage overhead, we utilize bitmaps as the index for material classifications. Here is the approach we employ:

(1) We encode each material classification to ensure that every classification has a unique cId.

(2) During the data upload process, we obtain the material classification cId to which the data belongs. We then extract keywords from the data, resulting in a keyword set W. We construct a corresponding bitmap for each element in the set (i.e., each keyword). We update the data cId in the bitmap by marking the cId-th bit as 1.

In the above step (2), an inverted index resembling keyword data is simultaneously constructed based on different cId values to maintain the metadata MPT. This means that the original data undergoes the aforementioned steps, resulting in the content depicted in Fig. 6.

2) Retrieval Integrity Verification: To ensure the integrity of the retrieved metadata set and maintain a verifiable retrieval index structure, the following processes should be implemented during the data upload and retrieval phases:

(1) Construction and Verification of Data Update Proof

To guarantee the completeness and accuracy of each updated data entry in the retrieval index, it is essential to construct a data update proof (VO_{update}) and verify it using the blockchain. This process ensures the consistency between $R_{(chain,cId)}$ maintained on the chain and the retrieval index stored in the cloud server.

The index update algorithm (Algorithm 1) is employed to update the retrieval index, while Algorithm 2 is used to construct VO_{update}. Before invoking the algorithm, a data structure called $proofDB$ should be constructed to store the node hash. $proofDB$ is implemented as a hash table, which supports key-value pair storage and essential operations such as adding, deleting, and querying. The algorithm takes the node's key, the MPT tree where the node is located, and a pointer to $proofDB$ (where the node hash is stored) as input parameters. The output is also a pointer to $proofDB$. The algorithm traverses from the root node, sequentially storing the encountered hash values along the node path, and finally outputs them.

The verification of the data update proof by blockchain parties can be divided into two steps:

① First, calculate the corresponding hash value based on the DataArr field in Table II provided by the web server. Then, compare this calculated hash value with DataHash to ensure that the newly added data is consistent with the data hash.

② Calculate the Merkle Patricia Tree (MPT) root hash based on the provided VO_{update} from the web server. Afterward, compare this calculated MPT root hash with RootHash to ensure the accuracy of VO_{update}.

Once both verification steps are successfully passed, the blockchain utilizes the built-in LevelDB to update $R_{(chain,cId)}$. In step ②, algorithm 3 is employed for validation. The input parameters for the algorithm include the MPT root hash and a pointer to $proofDB$ which stores the node hash. The output of the algorithm indicates whether the verification has passed.

(2) Construction and Verification of Data Integrity Proof

Algorithm 1 MPT update metadata set $Insert$

Input: MPT root node T, Keyword key, Serialized inserted $value$
Output: Insert completed subtree root node T'

if T is $empty$ **then**
  Create $leafN$, $Assign(N, key, value)$
  $T' \leftarrow N$
else if T is $branch$ **then**
  $T.child[key[0]] \leftarrow Insert(T.child[key[0]], key[1:], value)$
  $T' \leftarrow T$
else
  $p \leftarrow LongestCommonPrefix(T.key, key), l \leftarrow len(p)$
  if $l == len(T.key)$ **then**
    $Assign(T, key, value)$
    $T' \leftarrow T$
  else
    Create $branchB, extensionC$
    $B.child[D.key[l]] \leftarrow Insert(null, D.key[l+1:], D.value)$
    $B.child[key[l]] \leftarrow Insert(null, key[l+1:], value)$
    $Assign(C, key[:, l], B)$
    $T' \leftarrow C$
  end if
end if
return T'

When initiating a new data retrieval request, the web server initially needs to query the bitmap to retrieve the keyword associated with $cIdList$. Subsequently, Algorithm 4 is employed to obtain the metadata set. Algorithm 4 shares similarities with Algorithm 1 as it involves a recursive method. Prior to invocation, the keyword key to be searched must be processed into hexadecimal encoding. The input parameters for the algorithm include the current traversal node (initially the root node), the key to be searched (key), and the search pointer (initially set to 0). The output of the algorithm is the metadata collection.

After executing the data retrieval chain code on the blockchain side and receiving $Map < cId, R_{(chain,cId)} >$, the web server is required to utilize Algorithm 2 to construct VO_{cId} for each cId in the metadata set. This step involves obtaining $Map < cId, R_{(chain,cId)} >, VO_{cId}$, which is then forwarded to the client for verification. The verification algorithm used corresponds to Algorithm 3.

D. Multiparty Collaborative Service

In this section, we discuss multiparty collaborative service, and provide the framework and workflow of security multiparty computation scheme based on blockchain.

Given the complementary relationship among blockchain, federated learning, and secure multiparty computing, this paper adopts a blockchain-based solution to ensure the security of material data computation among multiple parties. It can ensure that each node has absolute control over its data, and all data calls can be audited in the whole process through the blockchain framework. Federated learning can realize that only the training model is shared among multiple participants rather than local raw data. Overall, the security problems in federated learning can be solved through secure multiparty computing, significantly reducing the risk of sensitive data leakage. It provides practical solutions for multiparty joint modeling, material properties prediction, and new materials generation on the material big-data sharing platform.

The workflow of a blockchain-based multiparty secure computing solution is shown in Fig. 7, which mainly includes the following four processes:

(1) Multiparty collaborative preparation

The data requestor (e.g., the high-temperature alloy material database) initially checks if the relevant results have been

Algorithm 2 Construct VO_{update}

Input: Keyword key, MPT root node T, $proofDB$
Output: $proofDB$

Initialize $nodes; hashList$
while $len(key) > 0$ and T is not $null$ **do**
  $N \leftarrow T$
  if N is $branch$ **then**
    $T \leftarrow N.child[key[0]], key \leftarrow key[1:]$
    $nodes.append(N)$
  else
    if $len(key) < len(N.key)$ or $N.key! = key[:len(n.key)]$ **then**
      $T \leftarrow null$
    else
      $T \leftarrow N.value; key \leftarrow key[len(N.key):]$
      $nodes.append(N)$
    end if
  end if
end while
for $node$ in $nodes$ **do**
  $hash \leftarrow nodeToHash(node)$
  $proofDB.update(hash, node.value)$
end for
return $proofDB$

Algorithm 3 Verify VO_{update}

Input: MPT RootHash $rootHash$, $proofDB$
Output: Verification results

while true **do**
  $hash \leftarrow proofDB.get(roofHash)$
  $node \leftarrow hashToNode(hash, rootHash, key)$
  if $node$ is $null$ **then**
    return $false$
  else if $node.value$ is not $null$ **then**
    return $true$
  else
    $rootHash \leftarrow node.child.hash$
  end if
end while

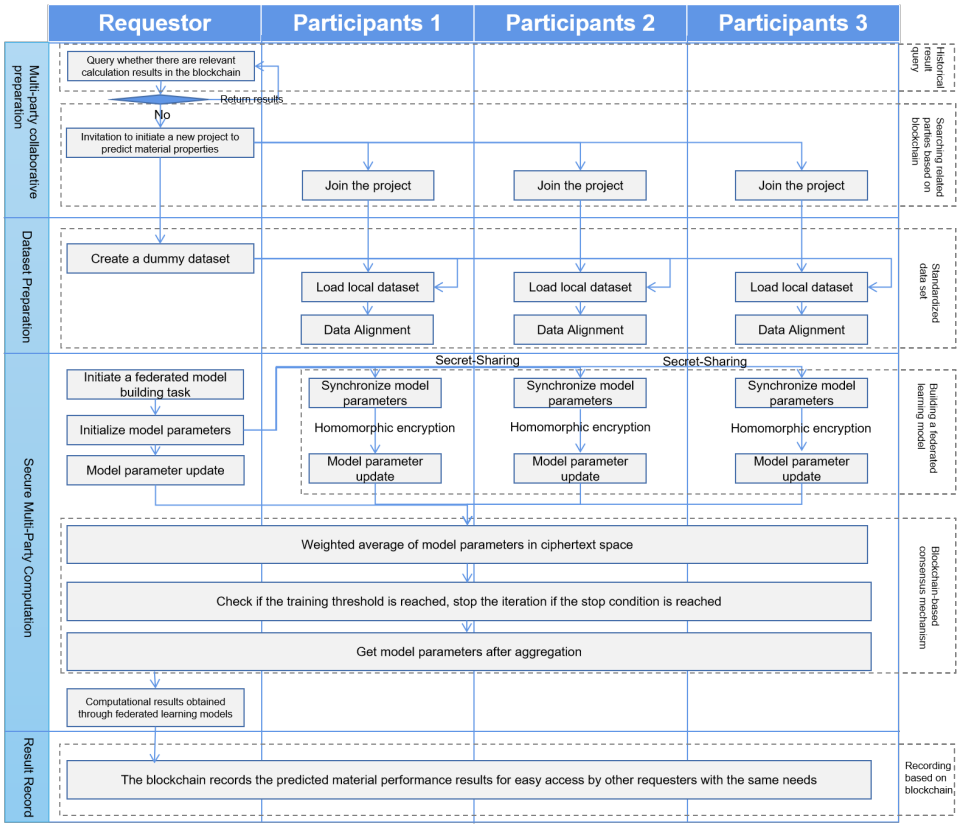


Fig. 7: Workflow diagram of security multiparty computation scheme based on blockchain.

Algorithm 4 Retrieval and get metadata set Get in MPT

Input: MPT root node T, Keyword key, Pointer pos
Output: Current lookup node N, Metadata set

if T is $empty$ **then**
  return $null, null$
else if T is $branch$ **then**
  $value, N \leftarrow Get(T.child[key[pos], key, pos + 1])$
  return $value, N$
else if T is $extension$ **then**
  $value, N \leftarrow Get(T.value, key, pos + len(T.key))$
  $T.value \leftarrow N$
else
  return $T, T.value$
end if

computed and stored on the blockchain. If no results are found, the requestor sends a project invitation to predict material performance on the platform. The project invitation is sent to the endorsement node of the alliance chain, which retrieves the relevant records stored on the blockchain, filters out the parties associated with the project, and broadcasts the project invitation to the identified parties. The selected parties become participants and consensus nodes in the alliance chain. Calculation contracts are negotiated with all involved parties prior to the multiparty computations. Calculation scripts containing model parameters initialized by federated learning, training models, calculation node parameters for the current task, node initial state, and execution conditions are deployed on the blockchain contract. The deployed contract can be automatically triggered and executed without human intervention.

(2) Preparation of participant datasets

To enable multiparty joint calculations, it is essential to standardize the characteristics and format of the input dataset between the demand side and each participating party. Therefore, the data requestor creates a virtual dataset and shares the dataset's features, formats, etc., with the participating parties. Each participant then loads their local data, references the data requestor's dataset, and aligns their data with the standardized dataset. This process ensures all participants have a consistent and compatible dataset for the multiparty calculations.

(3) Multiparty secure computing

Since all participants are relevant organizations in the field of materials, they share the same feature space but have different sample spaces. Hence, this paper proposes a horizontal federated learning model. Throughout the federated learning process, the consensus mechanism of the alliance chain is utilized for training the model parameter data, thereby effectively utilizing the nodes' computing resources.

The data requestor generates initialized model parameters based on the specific federated learning application scenario and synchronizes these initial parameters to other participants through the alliance chain. Secret-sharing techniques are employed during the parameter transmission process to ensure the security of model parameter transmission. The data requestor and each participant utilize the initial model parameters to up-

date the model locally. During the model update process, each party divides its local dataset into batches and computes the gradient for each data batch. The parameters are then updated based on the computed gradients. This process is repeated multiple times to obtain locally updated model parameters. Subsequently, each party performs homomorphic encryption on their local model parameters and shares them with other parties and the homomorphic public key. All parties receive the model parameter ciphertexts from other participants and perform weighted fusion in the ciphertext space, resulting in an aggregated model parameter ciphertext. The data requestor and all participants check whether the model parameters have converged using the blockchain consensus mechanism. If convergence is reached, the model training process is halted. Otherwise, the aggregated model parameter ciphertext is broadcasted to all participants who decrypt the ciphertext and update the model parameters again. This iterative process continues until all parties agree that the model has converged and the training process is not stopped. Finally, the trained model parameters are used to predict material performance.

Throughout the entire model training process, the model parameter data received by all parties remain in ciphertext form, and the model aggregation occurs in the ciphertext space. Consequently, it becomes difficult for any party to access the plaintext model parameters or derive the original data of other participants, thus ensuring the security and privacy of the original data.

(4) Result Recording

The data requestor uploads the calculation results to the alliance chain, allowing other requestors with similar calculation requirements to access the relevant result records. This approach helps in saving the overall calculation cost of the platform.

IV. Analysis and Discussion

Up to now, the portal website of the Secure-MBDF for MGED [29], [30] has collected more than 13 million pieces of valid material data. The platform's top five areas with the most data include special alloys, materials thermodynamics/kinetics, catalytic materials, first-principles calculations, and biomedical materials. Secure-MBDF offers comprehensive solutions for material data collection, storage, utilization, and data-sharing requirements. Participants can retrieve and calculate data, and data consumers from various fields can develop their research tools based on the service framework provided by Secure-MBDF to jointly predict material properties and develop new materials with other relevant parties.

A. Security Analysis

This section provides a security analysis of the material big data platform storage framework, focusing on the security of off-chain original data storage, security of on-chain transaction data storage, and integrity of retrieved data.

1) Security of off-chain original data storage: To ensure the protection of original data from potential leaks, the data owner encrypts the dataset using the AES algorithm before storing it on the cloud server. The encryption and decryption process utilize a symmetric key generated by the data owner. As a result, the cloud server can only access the encrypted version of the original data. This approach effectively mitigates the risk posed by an honest but curious cloud service provider and ensures the confidentiality of the original data. Additionally, the data owner maintains control over the actual data and retains authority over their data rights. During the data retrieval process, the data requester can only obtain the metadata of the original data from the blockchain. If the data requester wishes to access the original data, they must request permission from the data provider, retrieve the encrypted original data from the cloud server, and subsequently obtain the decryption key from the data owner through a key exchange protocol.

2) Security of on-chain transaction data storage: Blockchain, being a decentralized technology, provides resistance against security vulnerabilities arising from untrusted third parties. Utilizing blockchain to replace such entities, participants are interconnected through a distributed network. In the proposed blockchain-supported data storage scheme, the reliance on high-risk centralized trust, which often leads to data leakage, is eliminated. The data upload and retrieval transactions are automatically executed according to predetermined rules through smart contracts. Furthermore, transactions are stored on the blockchain in the form of hashes. Leveraging the uniqueness of the hash value, any modification made by an attacker to the transaction data will change the hash value. During the consensus process, nodes reject transaction data with inconsistent hash values, thereby ensuring the integrity of the data stored on the blockchain. Additionally, the entire history of data uploads and retrievals is transparently recorded in an immutable distributed ledger in the form of transactions, serving as verifiable evidence that can be publicly audited and traced.

3) Integrity of retrieved data: The integrity of data in the Merkle Patricia Trie (MPT) is maintained through the utilization of Merkle trees and hash values. The hash value of each non-leaf node in the MPT tree is computed based on the hash values of its child nodes. Whenever the value of a node changes, its corresponding hash value also changes, causing a cascade of hash value changes in its parent and ancestor nodes. This propagation of hash value changes extends along the path of the tree up to the root node. During the data upload process, the cloud server constructs a data update proof. If this proof is incorrect, indicating an error in the updated data of the retrieval index or the Merkle path hash, Algorithm 3 demonstrates that the verification results obtained on the blockchain will not match. Algorithm 2 establishes that forging an updated proof necessitates forging all hashes along the Merkle path. As mentioned earlier, the hash function exhibits collision resistance, making the probability of successfully forging such proofs extremely low, approximately 0.

B. Performance Analysis

In this subsection, we analyze the performance of our secured big-data sharing platform for materials genome engineering based on average response time and throughput. Firstly, we examine the platform's performance by integrating

TABLE III: Experimental equipment configuration.

Configuration	Parameters
CPU	Intel Xeon Platinum 8260 / 24 cores / 48 threads / 2.4GHz-3.9GHz
Operating System	Ubuntu 20.04
Number of Dockers	100
Memory	16*32GB
Language	Golang
Testing Tool	Apache Jmeter

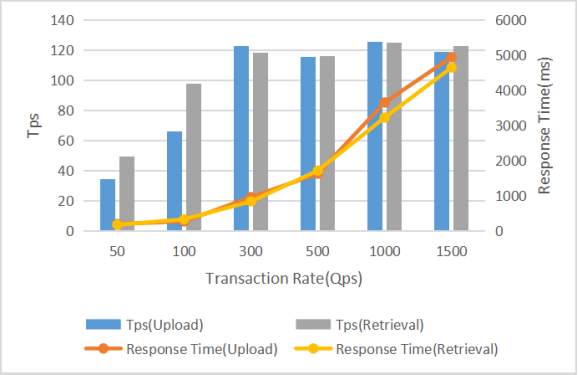


Fig. 8: Throughput and response time of the platform under different transaction rates.

our proposed blockchain framework to validate the impact of adopting the framework. Secondly, we evaluate the efficiency of our proposed retrieval algorithm to ensure that it enhances performance while maintaining adequate security measures. Throughput is defined as the number of transactions processed per second, while response time represents the average time required for each transaction to complete. The experimental results demonstrate that the utilization of the blockchain framework brings the platform's performance within an acceptable range, fully meeting the practical requirements of the system.

(1) The Enviroment Configuration

The experimental configuration environment needs to be considered from three aspects: the blockchain network, equipment configuration, and testing tools. In the experimental platform, we deployed 100 nodes to assess the impact of the number of participants on the platform's performance. Each node is launched as a Docker container for the blockchain network and connected to the Fabric network using Docker Swarm. The algorithms for data uploading, data retrieval, and model aggregation are all implemented as smart contracts by the material big data platform and submitted to the blockchain in the form of transactions. Additionally, the blockchain network configuration file defines network parameters such as organizations, peers, nodes, and channel names. Specific equipment configurations for our platform are presented in Table III. Apache JMeter was utilized as the testing tool to assess the platforms performance. The configuration files for these tools were set up with various transaction rates, transaction numbers, and workloads, including uploading and retrieval

It's important to note that the configuration parameters, such as the number of nodes, transaction rate, and transaction quantity used in the experiment, significantly exceed the requirements of real-world application scenarios. In actual scenarios [30], the number of nodes may be as low as 32, and the transaction rate typically won't exceed 100 transactions per second (tps). The platform can flexibly expand the node data according to the actual usage requirements. Therefore, our experiment configuration serves not only to test whether the platform's performance can meet the demands of real-world application scenarios but also to assess the platform's performance limits and scalability. In practical terms, this experiment provides insights into how well the platform can perform under extreme conditions and its potential for handling increased workloads, ensuring that it can adapt and scale effectively in response to the varying demands of different real-world applications

Five experimental cases were established to evaluate the critical performance indicators of the proposed platform:

1) *Case 1-The impact of blockchain on platforms:* In this case, we evaluate the impact of varying transaction rates on the platform's *uploading* and *retrieval* performance with the proposed blockchain framework. Several transaction rates were considered, specifically 50, 100, 300, 500, 1000, and 1500 transactions per second (tps).
2) *Case 2-The impact of new index structure on blockchain performance:* In this case, we assess the effect of adding our proposed index structure to the existing Hyperledger Fabric blockchain on the performance of data upload and retrieval. Similar transaction rates were considered, specifically 50, 100, 300, 500, 1000, and 1500 tps.
3) *Case 3-The parameter impact analysis:* In this case, a parameter impact analysis is conducted to evaluate the platform's performance across a range of parameter settings, including the number of participating nodes and the number of transactions. These analyses provide a more comprehensive assessment of our platform's strengths, helping us understand its robustness and identify optimal configurations for various scenarios.
4) *Case 4-The comparison with existing solutions:* In this comparison case, we aim to provide a more comprehensive evaluation by comparing the performance of our proposed method with other widely used and data-intensive material big data platforms, including AFLOW, COD, and OQMD. It's essential to note that not all platforms in the comparison set have data upload functionality, such as AFLOW, OQMD, and Materials Project. In the actual application process of the platform, data retrieval performance holds paramount importance. Therefore, our comparative analysis focuses solely on evaluating the performance of data retrieval.
5) *Case 5-The real-world use case:* In this scenario, three organizationsOrganization 1, 2, and 3collaborate to con-

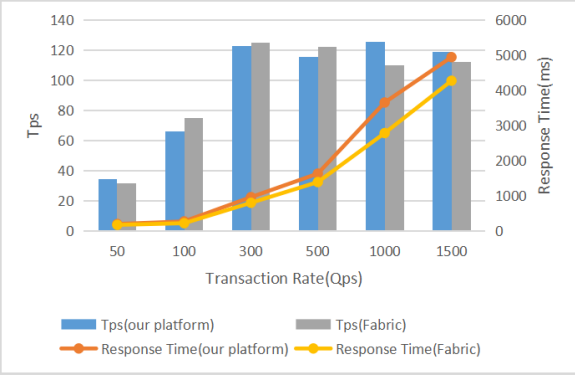


Fig. 9: The performance impact of adding optimized index structure at different transaction rates on data upload.

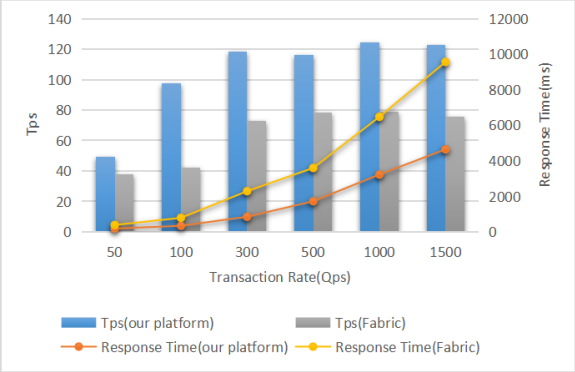


Fig. 10: The performance impact of adding optimized index structure at different transaction rates on data retrieval.

duct research on perovskite materials' performance. Each organization specializes in researching different types and uses of perovskite materials, resulting in distinct data sample sets. Despite these differences, all three organizations measure the same data features as required by their research projects. Their goal is to collaboratively predict the formation energy of perovskite materials without sharing their local data sets, aiming to enhance the model's training accuracy by leveraging their unique sample sets. In this collaborative setup, the three organizations collectively possess 4416 data points related to perovskite materials, with each organization's sample set comprising 20 distinct feature parameters. The objective of this case is to evaluate the accuracy and response time of collaborative prediction models implemented among these organizations. This evaluation is crucial as it enables us to assess the platform's effectiveness in practical industry applications.

(2) Results and Analysis

Case 1: Based on the experimental results, it has been observed that blockchain has a certain impact on the platform's performance, but within a reasonable range. For the upload function, the throughput stabilizes at around 120 transactions per second (tps), and as the throughput limit is reached, the response delay continues to increase. This implies that the specific environment can handle transactions at a rate of 120 tps, but it cannot fully reach 120 tps due to the limitations of the Hyperledger Fabric network's block-out strategy. The theoretical performance and maximum processing capacity of the system can be seen in Fig. 8 and are above 120 tps.

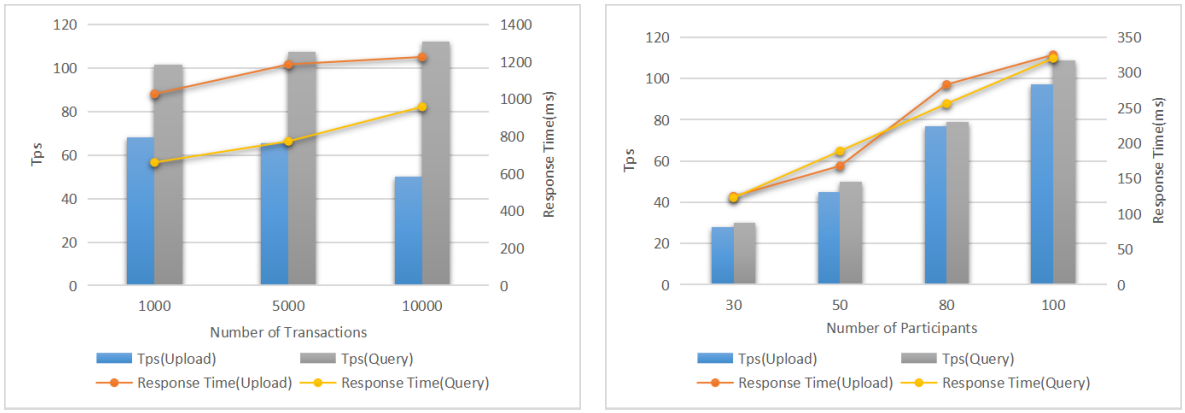
Similarly, the throughput of the retrieval function remains stable at around 120 tps, and as the transaction rate increases, the response delay continues to increase. This indicates that the specific environment can handle transactions at a rate of 120 tps. However, when the transaction rate surpasses this limit, the block-out condition is quickly reached, resulting in the overall tps of the system becoming higher and higher. The maximum processing capacity of the system for retrieval is shown in Fig. 8 and is above 120 tps.

Case 2: As shown in Fig. 9, during the upload phase, the platform's throughput and latency are not significantly affected when the transaction rate is below 500 tps, compared to the platform without the optimized index structure. The impact of the index structure on the platform's performance becomes noticeable when the transaction rate exceeds 500 tps. At transaction rates higher than 500, our platform exhibits slightly higher response times than Fabric due to the additional verification operations required for data update proof in our blockchain network. However, in practical usage scenarios, the transaction rate generally does not exceed 200 tps [30]. Hence, the impact of the transaction rate on the platform is considered acceptable after incorporating our proposed index structure. Additionally, as the number of concurrent transactions increases, enhancing hardware resources and network bandwidth can easily improve throughput and latency performance without adjustments to the system architecture and services.

During the retrieval phase, as depicted in Fig. 10, the blockchain network demonstrates the capability to handle 100 tps without noticeable delays. When the transaction rate reaches 100 tps, the response time for data retrieval is a mere 338 milliseconds. This response time is nearly imperceptible to users, indicating the efficiency of our retrieval process. Moreover, our proposed method exhibits a substantial decrease in average retrieval time compared to the original retrieval methods of Fabric. These results validate the enhanced retrieval performance achieved through the adoption of the bitmap and MPT-based retrieval method in a blockchain environment.

Case 3: In Fig. 11-(a), the platform's storage and retrieval performance were evaluated by varying the total number of transactions, specifically 1,000, 5,000, and 10,000 transactions. These transactions were sent to the blockchain at a rate of 120 queries per second (qps), a throughput limit established in the previous experiment. Fig. 11-(a) illustrates the average throughput and latency for both "upload" and "retrieve" transaction types concerning the number of transactions. Theoretically, utilizing a transaction rate that reaches the throughput limit to test this scenario should not significantly impact throughput and latency. The experimental results indeed validate this theoretical analysis. During data upload and retrieval processes, the total number of transactions minimally influences throughput and latency, with latency differences of only a few tens of milliseconds

Fig. 11-(b) investigates the impact of the number of participants on the storage and retrieval performance of the



(a) (b)

Fig. 11: The performance comparison of data storage and retrieval under the influence of different parameters (a) number of transactions (b) number of participants.

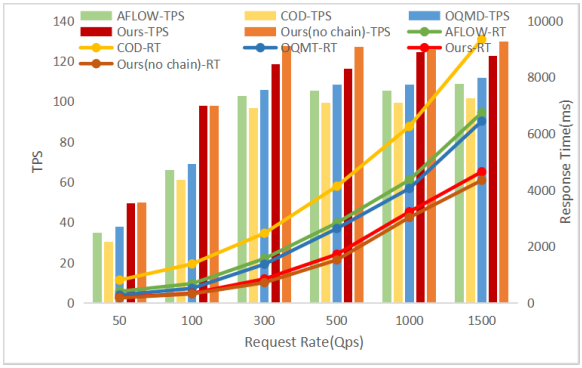


Fig. 12: The performance comparison chart of various material big data platforms under different request rates.

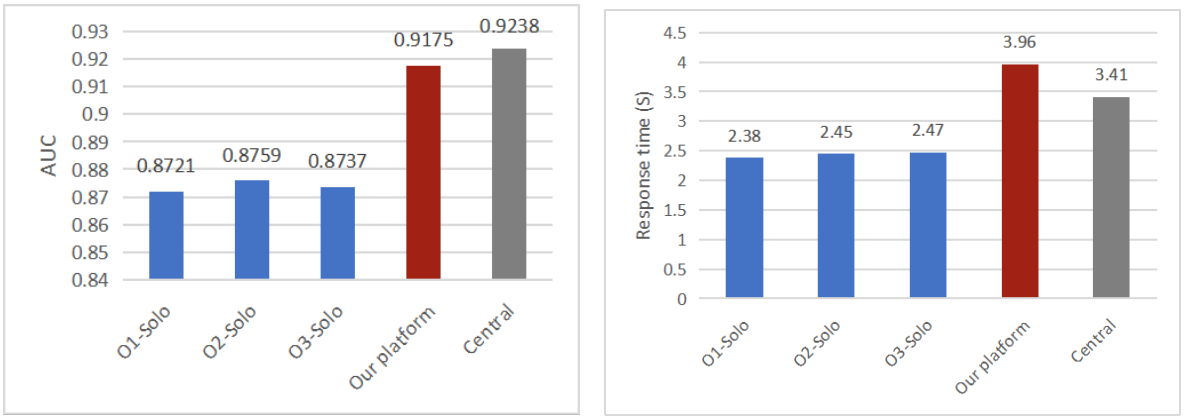
platform. We varied the number of participants to 30, 50, 80, and 100. Each participant sent requests to the blockchain simultaneously to assess the throughput and latency of data upload and retrieval transactions. According to the experimental results, both throughput and latency increased as the number of participants grew during data upload and retrieval processes. In fact, when 100 users simultaneously sent transactions, an improvement in throughput was observed compared to the scenario where a single user sent transactions at a rate of 100 queries per second (qps).

This improvement occurred because when 100 users simultaneously sent transactions, the system had to process multiple concurrent requests. Through parallel processing, the system could handle multiple requests simultaneously without waiting for the previous request to complete, thereby increasing throughput. In contrast, when a single user sent transactions at a rate of 100 qps, the requests might be continuous, with less apparent concurrency. This continuous flow could lead to more queuing and waiting during request handling, potentially reducing throughput. However, the primary reason for increased response time in data upload and retrieval with a growing number of participants is the heightened system load at any given moment. This increased load forces the system to concurrently handle a larger volume of tasks, resulting in longer response time.

Case 4: Fig. 12 compares the response time and throughput of different big data platforms under various request rates. In the figure, RT represents response time, TPS represents throughput, and Ours(no chain) represents the platform before the deployment of the blockchain framework. The data extracted from the graph indicates that this platform outperforms others in terms of retrieval throughput and exhibits a more pronounced advantage in response time for data retrieval compared to other platforms. When comparing this platform to its state before deploying a blockchain framework, there is a slight decrease in throughput and an increase in response time, albeit on a millisecond scale. The primary reason for this is that while the blockchain framework with improved indexing structures enhances data retrieval efficiency, it also increases the communication frequency between nodes during data retrieval transactions and data integrity verification processes, thereby impacting the platform's retrieval performance. Despite the minor decrease in performance, the platform still meets application requirements during practical usage. For users, the millisecond-level difference is nearly imperceptible.

Simultaneously, a slight loss in retrieval performance on the platform brings two significant benefits. Firstly, the blockchain stores structured metadata corresponding to heterogeneous raw data from various sources. Given the mapping relationship between on-chain data and off-chain heterogeneous raw data, we utilize on-chain structured metadata to retrieve and manage the platform's heterogeneous data successfully. This solution effectively addresses the challenges of low efficiency in retrieving heterogeneous data from various sources and managing data storage. Secondly, the platform also validates the integrity of data during retrieval, ensuring that the retrieved raw data has not been tampered with, which is a security concern that other big data platforms have not considered.

Case 5: In our tests, we examined the runtime and model prediction accuracy of the federated learning model under the platform framework, referred to as the "Platform Model." We also compared it with local predictive models and centralized training predictive models independently performed



(a) (b)

Fig. 13: The performance comparison diagram of federated learning methods in perovskite performance prediction scenario (a) response time (b) model accuracy.

by each organization. From Fig. 13-(a), it can be observed that the response time of the local predictive models from different organizations are essentially the same. Meanwhile, the platform conducts distributed federated learning model training based on a blockchain architecture, incurring some communication time costs among organizations, leading to slightly longer runtime for the Platform Model compared to the local predictive models and the centralized predictive model. In comparison to the centralized model, the average response time of the Platform Model only increased by 0.55 seconds. However, in terms of actual user experience, these differences are almost imperceptible and well within an acceptable range.

Fig. 13-(b) clearly shows that the Platform Model significantly improves prediction accuracy while preserving the privacy of local datasets from various organizations compared to the local predictive model. However, when compared to the centralized predictive model, the accuracy of the Platform Model experiences a slight decrease. This is primarily because, in the Platform Model, organizations do not expose their raw datasets to other participants, leading to some information loss to some extent. Nevertheless, the data privacy of organizations is protected without a significant decline in performance.

V. Conclusion and Prospect

The secured big-data sharing platform for materials genome engineering serves as a national science and technology infrastructure platform. It leverages this platform to publish and provide services through its portal website, supporting material selection and expediting material design and optimization. This contribution is of significant importance to national economic development. The Secure-MBDF, proposed in this paper, employs blockchain, federated learning, secure multiparty computing, and other technologies to realize the secure sharing of data, overcoming the issue of "data islands" in material information. Its systematic and scientific nature drives the advancement of materials genome engineering. In the era of big data, material data plays an increasingly vital role, facilitating the profound development of material science innovation. The integration of material data with information and other fields presents substantial challenges to material data researchers. The analysis and mining services provided by Secure-MBDF will expedite the progression of the fourth paradigm, data-driven material research, and development.

In future research, the Secure-MBDF will be further enhanced and optimized based on the existing architecture and related solutions. For instance, we may explore the combination of trusted execution environments with blockchain technology to ensure the confidentiality and integrity of programs and data during execution. This enhancement will further bolster the security of our platform during data sharing.

References

[1] J. Wan, S. Tang, Q. Hua, D. Li, C. Liu, J. Lloret, Context-aware cloud robotics for material handling in cognitive industrial internet of things, IEEE Internet of Things Journal 5 (4) (2017) 2272–2281.

[2] R. Ashima, A. Haleem, S. Bahl, M. Javaid, S. K. Mahla, S. Singh, Automation and manufacturing of smart materials in additive manufacturing technologies using internet of things towards the adoption of industry 4.0, Materials Today: Proceedings 45 (2021) 5081–5088.

[3] A. Agrawal, A. Choudhary, Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science, APL Materials 4 (5) (2016) 053208.

[4] J. J. de Pablo, N. E. Jackson, M. A. Webb, L.-Q. Chen, J. E. Moore, D. Morgan, R. Jacobs, T. Pollock, D. G. Schlom, E. S. Toberer, J. Analytis, I. Dabo, D. M. DeLongchamp, G. A. Fiete, G. M. Grason, G. Hautier, Y. Mo, K. Rajan, E. J. Reed, E. Rodriguez, V. Stevanovic, J. Suntivich, K. Thornton, J.-C. Zhao, New frontiers for the materials genome initiative, npj Computational Materials 5 (1) (2019) 41.

[5] S. Suhail, R. Hussain, R. Jurdak, C. S. Hong, Trustworthy digital twins in the industrial internet of things with blockchain, IEEE Internet Computing.

[6] Aflow - Automatic FLOW for Materials Discovery. URL https://aflowlib.org/

[7] S. P. Huber, S. Zoupanos, M. Uhrin, L. Talirz, L. Kahle, R. Häuselmann, D. Gresch, T. Müller, A. V. Yakutovich, C. W. Andersen, F. F. Ramirez, C. S. Adorf, F. Gargiulo, S. Kumbhar, E. Passaro, C. Johnston, A. Merkys, A. Cepellotti, N. Mounet, N. Marzari, B. Kozinsky, G. Pizzi, AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance, Scientific Data 7 (1) (2020) 300.

[8] C. Draxl, M. Scheffler, NOMAD: The FAIR concept for big data-driven materials science, MRS Bulletin 43 (9) (2018) 676–682. doi:10.1557/mrs.2018.208.

[9] S. R. Pokhrel, J. Choi, Federated Learning With Blockchain for Autonomous Vehicles: Analysis and Design Challenges, IEEE Transactions on Communications 68 (8) (2020) 4734–4746.

[10] C. H. Liu, Q. Lin, S. Wen, Blockchain-Enabled Data Collection and Sharing for Industrial IoT With Deep Reinforcement Learning, IEEE Transactions on Industrial Informatics 15 (6) (2019) 3516–3526.
[11] Z. Chen, W. Xu, B. Wang, H. Yu, A blockchain-based preserving and sharing system for medical data privacy, Future Generation Computer Systems 124 (2021) 338–350.
[12] L. Himanen, A. Geurts, A. S. Foster, P. Rinke, Data-Driven Materials Science: Status, Challenges, and Perspectives, Advanced Science 6 (21) (2019) 1900808.
[13] Crystallography Open Database.
URL http://www.crystallography.net/cod/
[14] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, I. Foster, The Materials Data Facility: Data Services to Advance Materials Science Research, JOM 68.
[15] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K. A. Persson, Commentary: The Materials Project: A materials genome approach to accelerating materials innovation, APL Materials 1 (1) (2013) 011002.
[16] S. Curtarolo, W. Setyawan, G. L. W. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, D. Morgan, AFLOW: An automatic framework for high-throughput materials discovery, Computational Materials Science 58 (2012) 218–226, arXiv: 1308.5715.
[17] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R. Taylor, L. Nelson, G. Hart, S. Sanvito, M. Buongiorno Nardelli, N. Mingo, O. Levy, AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations, Computational Materials Science 58 (2012) 227–235.
[18] S. Graulis, A. Dakevi, A. Merkys, D. Chateigner, L. Lutterotti, M. Quirós, N. R. Serebryanaya, P. Moeck, R. T. Downs, A. Le Bail, Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration, Nucleic Acids Research 40 (D1) (2012) D420–D427.
[19] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton, Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD), JOM 65 (11) (2013) 1501–1509.
[20] Open Materials Database.
URL https://openmaterialsdb.se/
[21] Atsteel.
URL https://www.atsteel.com.cn/
[22] N. Deepa, Q.-V. Pham, D. C. Nguyen, S. Bhattacharya, B. Prabadevi, T. R. Gadekallu, P. K. R. Maddikunta, F. Fang, P. N. Pathirana, A survey on blockchain for big data: Approaches, opportunities, and future directions, Future Generation Computer Systems 131 209–226.
[23] J. Yang, J. Wen, B. Jiang, H. Wang, Blockchain-Based Sharing and Tamper-Proof Framework of Big Data Networking, IEEE Network 34 (4) (2020) 62–67.
[24] Y. Yang, L. Wei, J. Wu, C. Long, Block-SMPC: A Blockchain-based Secure Multi-party Computation for Privacy-Protected Data Sharing, in: Proceedings of the 2020 The 2nd International Conference on Blockchain Technology, ACM, Hilo HI USA, 2020, pp. 46–51.
[25] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, H. V. Poor, Federated learning meets blockchain in edge computing: Opportunities and challenges, IEEE Internet of Things Journal 8 (16) 12806–12825.
[26] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, H. V. Poor, Federated Learning With Differential Privacy: Algorithms and Performance Analysis, IEEE Transactions on Information Forensics and Security 15 (2020) 3454–3469.
[27] H. Chai, S. Leng, Y. Chen, K. Zhang, A Hierarchical Blockchain-Enabled Federated Learning Algorithm for Knowledge Sharing in Internet of Vehicles, IEEE Transactions on Intelligent Transportation Systems 22 (7) (2021) 3975–3986.
[28] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukoli, S. W. Cocco, J. Yellick, Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, ACM, Porto Portugal, 2018, pp. 1–15.
[29] S. Liu, Y. Su, H. Yin, D. Zhang, J. He, H. Huang, X. Jiang, X. Wang, H. Gong, Z. Li, H. Xiu, J. Wan, X. Zhang, An infrastructure with user-centered presentation data model for integrated management of materials data and services, npj Computational Materials 7 (1) (2021) 88.
[30] National meterial data management & service.
URL http://mged.nmdms.ustb.edu.cn/analytics/

Ran Wang received the B.E. degree from the Beijing Information Science and Technology University, China in 2013, and the M.S. degree from the University of Science and Technology Beijing (USTB), China in 2016. She is currently working toward the Doctoral degree at University of Science and Technology Beijing. Her research interests include distributed security, blockchain and internet of things.

Cheng Xu received the B.E., M.S. and Ph.D. degree from the University of Science and Technology Beijing (USTB), China in 2012, 2015 and 2019 respectively. He is currently an Associate Professor with School of Computer and Communication Engineering, University of Science and Technology Beijing. He is supported by the Post-doctoral Innovative Talent Support Program from Chinese government in 2019. His current research interests include swarm intelligence, multi-agent reinforcement learning, distributed security and internet of things. He is an Associate Editor of International Journal of Wireless Information Networks, and a member of the IEEE.

Fangwen Ye is currently working toward the Master degree at University of Science and Technology Beijing. His research interests include distributed security, multi-modal navigation and internet of things.

Sisui Tang is currently working toward the Master degree at University of Science and Technology Beijing. Her research interests include distributed security, patter recognition and internet of things.

Xiaotong Zhang is currently a Professor with School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include material big-data, database systems and internet of things.