Automation in Construction 167 (2024) 105720

Contents lists available at ScienceDirect

Automation in Construction

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

Consortium blockchain-based tunnel data bank for traceable sharing and treatment of structural health monitoring data

Dong-Ming Zhang [a], Cong Nie [a], Jin-Zhang Zhang [a,b,*], Hong-Wei Huang [a,*], Xu Huang [c]

[a] Key Laboratory of Geotechnical and Underground Engineering of Ministry of Education and Department of Geotechnical Engineering, Tongji University, Shanghai 200092, China

[b] Department of Civil and Environmental Engineering, Hong Kong University of Science and Technology, Hong Kong SAR, China

[c] China Railway Eryuan East China Investigation and Design Institute Co.Ltd, Zhejiang 239001, China

ARTICLE INFO

Keywords:
Tunnel structural health monitoring
Data sharing
Consortium blockchain
Smart contract
Hyperledger fabric

ABSTRACT

The volume of structural health monitoring (SHM) data is rapidly increasing due to the explosive growth of information from constructed infrastructures. However, data storage by different monitoring organizations is often isolated, even when they serve the same construction project, resulting in "data silos". Improving the efficiency of sharing massive monitoring data and developing a reliable, distributed system can significantly enhance data mining efficiency. In this paper, a secure and anti-tampering data bank for sharing tunnel SHM data was innovatively constructed based on blockchain technology, significantly enhancing traceable storage, regulation, and collaboration between different organizations. A smart contract was implemented to manage heterogeneous data storage, monitor incremental calculations, and make alert judgments. A series of tests assessing traceability and applicability were conducted using monitoring data from a tunnel in Hangzhou, collected through Wireless Sensor Networks. The results demonstrate that the constructed data bank effectively facilitates data exchange and sharing among multiple monitoring parties.

1. Introduction

The health monitoring of tunnel structures is crucial for ensuring the normal functioning of cities, particularly in transportation [1,2]. The pace of urbanization in China has accelerated significantly over the past few decades, leading to the construction and utilization of extensive infrastructure. Urban rail transit systems are vital components of transportation networks in metropolitan areas. By the end of 2022, 55 cities in China had metro systems with a combined length of 10,287.45 km, including 308 urban transit lines [3]. Structural health monitoring (SHM) of tunnels is a critical aspect of operational safety. Among current monitoring methods, point clouds and images represent substantial volumes of refined monitoring data. For instance, monitoring just one kilometer generates several gigabytes of image data, and monitoring 10,000 km can produce tens of terabytes. The variety of data formats adds to the challenge. Without proper control over storage formats and paths, accurately assessing the true state of a structure can be difficult, and subtle data tampering might go undetected, posing significant security risks. Therefore, effective risk perception and timely communication feedback are crucial, and they are a major development focus for large asset management, such as risk management for metro tunnel systems in China.

Various methods exist for tunnel SHM, each corresponding to a different data structure. Traditional monitoring of longitudinal settlement primarily relies on manual methods using leveling or total stations [4], with findings documented in monitoring reports. Additionally, several automatic methods have been proposed for real-time settlement monitoring, including static-level systems and robotic total stations [5,6]. The static system detects differential settlement by measuring changes in the liquid level within containers. Equipped with data transmission instruments, the static-level system can automatically provide monitoring data. Furthermore, Wireless Sensor Networks (WSN) offer flexible approaches to data collection, transmission, and processing. WSN establishes a distributed multi-hop ad hoc network that wirelessly connects sensors to the internet through commercial signals [7–9]. WSN instruments can transmit data to cloud data banks, where it can be further processed into structured data using a well-organized JSON format. Certain devices, such as tunnel scanners [10], are

* Corresponding authors at: Key Laboratory of Geotechnical and Underground Engineering of Ministry of Education and Department of Geotechnical Engineering, Tongji University, Shanghai 200092, China.

E-mail addresses: zhangjz@tongji.edu.cn (J.-Z. Zhang), huanghw@tongji.edu.cn (H.-W. Huang).

https://doi.org/10.1016/j.autcon.2024.105720
Received 23 September 2023; Received in revised form 15 August 2024; Accepted 16 August 2024
Available online 25 August 2024

capable of collecting large amounts of data in a single scan, which is subsequently transferred manually.

Tunnel SHM generates vast and diverse heterogeneous data. Inadequate data management can lead to disorder, significantly reducing the data's usefulness. Rational data exchange and sharing among different organizations can unlock additional value. Existing data management and sharing systems are typically centralized, with data stored in centralized databases. However, this isolated storage presents the risk of a single point of failure, compromising availability and security in the event of a failure or attack on the centralized server. Moreover, tracking data modifications can be challenging, requiring complex traceability mechanisms to achieve tamper-proof functionality, which can lack transparency and traceability. Additionally, traditional systems may encounter consistency issues due to isolated storage, particularly when dealing with heterogeneous data from multiple sources. Finally, traditional storage may not promptly reflect the most recent status of the data due to synchronization latency.

To address these drawbacks in existing SHM systems, this paper introduces blockchain technology to implement a traceable and secure tunnel SHM data management and sharing system, called a Tunnel Data Bank (TDB). The origin of blockchain can be traced back to 2008 when an individual or group using the pseudonym "Satoshi Nakamoto" released a white paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" [11]. This white paper proposed a digital currency system based on a decentralized consensus mechanism and cryptographic modeling technologies. In the subsequent years, blockchain technology experienced rapid development alongside Bitcoin, with other types of blockchain, such as Ethereum [12] and Hyperledger [13] also emerging. The development of blockchain has progressed through three distinct phases. The first phase is the single-chain blockchain, represented by Bitcoin, which is mainly used for the issuance and trading of digital currency. The second phase is the multi-field blockchain, exemplified by Ethereum and Hyperledger, which supports not only digital currency transactions but also smart contracts, digital identity authentication, the Internet of Things (IoT), and other application scenarios [14]. The third phase is the federated and private chain blockchain, primarily intended for sharing within enterprises and institutions, providing higher security and controllability [15]. In this phase, blockchain has been widely applied in the industrial sector, inspiring significant work in various areas of civil engineering.

Significant efforts have been made in construction management, structural health monitoring (SHM), and the digitization of Building Information Modeling (BIM), offering valuable insights into these fields. The integration of blockchain technology can be applied throughout the entire lifecycle of construction projects. The feasibility of using both public and private blockchain technologies in the construction industry has also been explored [16]. Blockchain frameworks that integrate digital twins for traceable data communication have been developed [17,18], and blockchain technology has been implemented to enhance precast supply chains [19]. As a digital tool in construction, BIM inherently offers advantages for integration with blockchain technology. Blockchain BIM as a service (BaaS) has been developed within a permissioned blockchain to facilitate the interoperability of information, semantics, and meaningful inferences [20]. The integration of blockchain and BIM can benefit construction supply chains [21]. To mitigate the risks associated with the centralized architecture of BIM production environments and data storage, several studies have focused on enhancing the distributed, reliable, and traceable aspects through the integration of blockchain with BIM [22–24].

In the context of IoT-based SHM, optimizing data management can be achieved through the utilization of blockchain's distributed storage. The application of blockchain technology, in the form of a public chain, was initially introduced for health monitoring of underground engineering structures [25], offering significant inspirational value despite not optimizing energy consumption resulting from the consensus algorithm. Additionally, more intricately designed blockchain-distributed systems have been proposed for SHM applications [26]. Simultaneously, a specifically tailored system integrating blockchain with IoT devices has been developed [27,28], alongside ongoing research on optimizing 5G networks [29] and data transmission [30].

Previous studies have offered valuable insights into edge devices, data transmission methods, and data management frameworks. However, a secure and traceable data bank explicitly designed for tunnel SHM within a consortium blockchain has not yet been proposed, despite the urgent need for such a solution to effectively manage the massive and heterogeneous data in tunnel SHM. For instance, safeguarding data privacy requires the implementation of an access control system for data acquisition, unlike the transaction transparency observed in public blockchains. Additionally, the widely used consensus algorithms, known for their significant energy consumption, demand the development of lightweight and environmentally friendly alternatives. Finally, addressing the diverse and heterogeneous tunnel SHM data calls for the meticulous design of smart contracts and storage patterns. Therefore, it is crucial to conduct research on a tailored tunnel data repository based on consortium blockchains.

The objective of this paper is to explore the potential applications of consortium blockchain technology in civil engineering, with a specific focus on sharing tunnel structural health monitoring data. A consortium blockchain was established to create a data-sharing repository for tunnel SHM, enabling efficient data exchange among multiple monitoring parties while ensuring privacy and security. Compared to existing blockchain-based SHM systems, this paper presents the following key contributions: (1) The introduction of a consortium blockchain framework tailored for tunnel SHM, which facilitates distributed storage, traceability, security, and fair sharing of heterogeneous data from multiple sources. (2) The proposal of a smart contract for managing heterogeneous data, enabling permission management, modification traceability, incremental computation, and risk warnings for tunnel SHM data. (3) The adoption of consensus algorithms with minimal energy consumption to achieve computationally and environmentally friendly solutions, assuming mutual trust among consortium organizations.

The paper is organized as follows: First, the context of tunnel SHM and the origin and development of blockchain technology are discussed. Next, detailed information on the data structure, consensus mechanism, and smart contract of blockchain is provided, with a particular focus on consortium blockchains, specifically the application of Hyperledger Fabric. The third section proposes a framework for a data-sharing repository dedicated to tunnel SHM, based on a consortium blockchain. This includes a smart contract designed for multi-source heterogeneous data storage, data structures for storing such data, and logic for queries and warnings. Additionally, the paper details the interaction logic for computing increments based on historical data within the smart contract. Following this, the management of multi-source heterogeneous data for tunnel SHM within the data repository is introduced. Finally, a specific application of the proposed data repository is demonstrated, showcasing the upload of monitoring data from the Hangzhou Metro, collected by the WSN, to the established data-sharing repository. Concurrently, service delay and stability tests are conducted. This research highlights the effectiveness and broad potential of the proposed blockchain-based data repository.

2. Blockchain technology

As a novel type of distributed technology, blockchain is a decentralized network architecture consisting of nodes in a blockchain network based on a peer-to-peer (P2P) network, without a centralized authority or manager controlling all nodes. The decentralized nature of blockchain is rooted in a series of advanced technologies such as P2P protocols [31], robust computational algorithms, and data encryption [32]. It introduces a trust mechanism based on decentralization into economic, knowledge, and social systems, achieving a power balance among the nodes participating in the network through consensus mechanisms.

2.1. Data structure of blockchain

The data structure of a blockchain is based on a block that serves as a fundamental unit for storage and transmission. Each block contains the current transaction details including a hash value that points to the previous and present blocks. In a blockchain structure, all the blocks are interconnected through hash pointers to form a directed acyclic graph data structure, as shown in Fig. 1. This configuration enables the blockchain to not only store transaction information, but also ensure the immutability and integrity of the data, thereby mitigating the risk of collusion attacks.

Each block consists of a block header and body. The block header encapsulates information, such as the height of the block, hash value of the current block, hash pointer to the previous block, and assorted network parameters. Simultaneously, the body of the block discloses all transaction records embedded in it. In the cryptocurrency domain, each transaction requires signing and timestamping for inclusion in a block. Encryption algorithms play a pivotal role in ensuring transactions and user privacy. Zero knowledge-proof technology can authentically verify certain facts without disclosing information, thereby safeguarding user privacy [33].

For the swift and efficient verification of data integrity, each block in the blockchain utilizes the Merkle tree as its data storage structure. Proposed by the computer scientist Ralph Merkle in 1979, the Merkle tree has become the cornerstone of blockchain technology [34]. The root node stores the hash values of all data blocks, and the intermediate nodes are derived from the hash values of adjacent data blocks through hash functions, as shown in Fig. 2. This efficiency arises from the fact that only the hash values of the upper-level nodes require recalculation. Incorporated into the header of each block, a Merkle root hash value is used to authenticate the integrity of all transactions within the block. Upon receiving a new block, a node can validate the integrity of all the transactions by comparing its Merkle root hash value with that of the preceding block. This process guarantees transaction order and timestamp accuracy, and safeguards against data tampering and forgery. The Merkle tree structure plays a significant role in blockchain technology, providing a substantial assurance of blockchain security and trustworthiness.

2.2. Consensus mechanism of blockchain

The consensus mechanism in a blockchain refers to the process by which participants reach an agreement through a certain algorithm in a distributed system. Successful consensus mechanisms must be fair, secure, open and efficient. This paper reviewed the various consensus mechanisms of blockchain technology. 1) Proof of Work (PoW): The earliest consensus mechanism was employed by Bitcoin [35]. Miners solve complex problems in order to verify transactions and receive cryptocurrency rewards. The PoW is suitable for public chains with high energy consumption. 2) Proof of Stake (PoS): An enhanced version of PoW has been gradually adopted by various blockchain projects [36]. Participants lock a certain amount of encrypted digital currency as a voting right to verify their transactions. Holding equity, which increases over time, gives participants greater rights to transaction verification. 3) Proof of Authority (PoA): A centralized consensus mechanism in which authorized entities verify transactions and issue new blocks [37]. Although it is effective against tampering and fraud, it is susceptible to authority manipulation. 4) Delegated Proof of Stake (DPoS): Similar to the PoS, holders delegate voting rights to representatives for transaction verification [38]. Representatives elected based on authorized holdings verify the transactions. 5) Byzantine Fault Tolerance (BFT): This Addresses the Byzantine General problem, involving malicious nodes that disrupt the system's correctness [39]. The nodes achieve consensus through dialogue protocols and mutual confirmations. BFT is widely used in consortia and private chains because of its fast transaction confirmation and high fault tolerance. This is particularly suitable for scenarios requiring rapid transaction confirmation and system reliability. 6) Raft: Term-based consensus ensures consistent state machines, safety, and liveness [40]. Raft is suitable for consortium blockchains that operate under the assumption of mutual trust among organizations. Consequently, it reduces the need for extensive measures to prevent malicious attacks by focusing on fault tolerance. Consequently, Raft is environmentally friendly and energy efficient. This paper proposes a consortium blockchain utilizing the Raft consensus mechanism to ensure fast response, simple implementation, strong fault tolerance, and low energy consumption.

2.3. Smart contract of blockchain

A smart contract is an automated contract executed on a blockchain. It consists of definitions of rules and conditions for contract participants. By leveraging the distributed and tamper-resistant features of blockchain technology, a smart contract ensures transparent and reliable contract execution. The smart contracts utilized in this paper are distributed across organizations within the consortium blockchain. It features intricate logic, well-defined permission control, and effective traceability.

2.4. Consortium blockchain and hyperledger fabric

A consortium blockchain is designed to establish a shared and controlled blockchain network among specific organizations [41]. Unlike public blockchains, consortium blockchains consist of predetermined participants, who are usually entities with shared interests. Consortium blockchains provide easy-to-manage permission controls and convenient data sharing, and surpass public chains in terms of response speed and distributed storage consistency. The Hyperledger Fabric, developed by the Linux Foundation, is one of the most notable works in this field. This is an open-source consortium blockchain

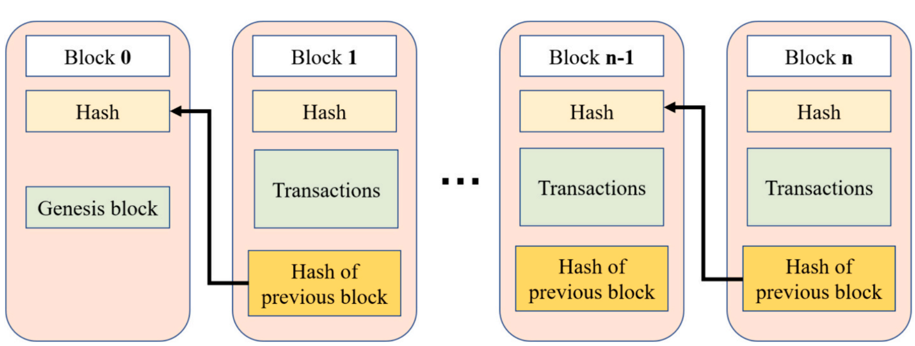


Fig. 1. Data structure of the blockchain.

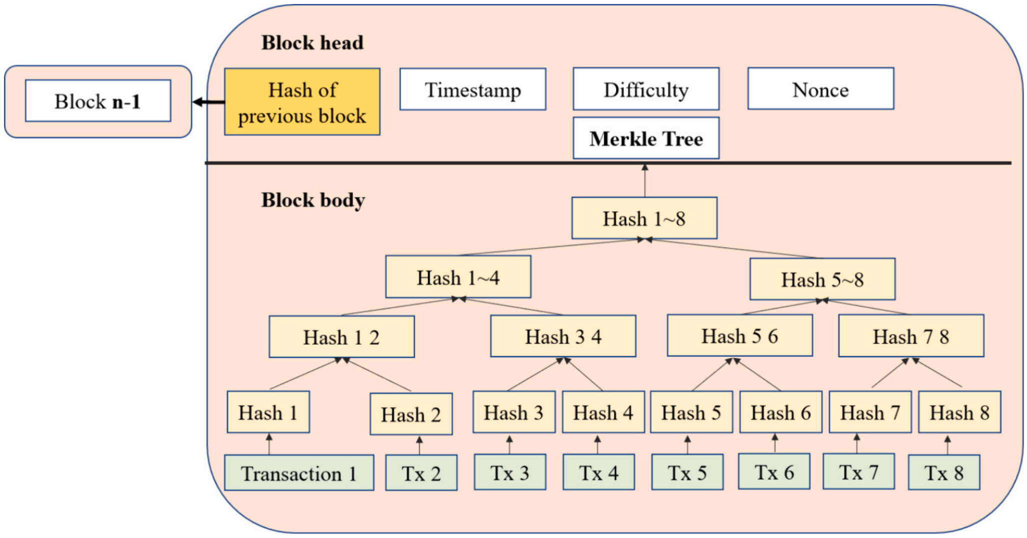


Fig. 2. Structure of a Merkle tree in a block.

framework specifically designed to support the deployment of enterprise-level blockchain applications. Hyperledger Fabric has a modular design, supports flexible identity management and permission control, accommodates complex smart contract logic, and exhibits robust fault tolerance constraints. Considering these advantages, the proposed tunnel data bank adopted Hyperledger Fabric as its foundational framework.

3. Proposed traceable tunnel data bank

3.1. Framework of the traceable data bank

Tunnels often face disruptions due to adjacent construction activities and uncertain strata during their operational phase [42–47,50]. In this context, monitoring organizations are responsible for overseeing existing tunnels, construction organizations ensure construction safety, supervising organizations oversee the process, and owner organizations coordinate overall activities. This means that monitoring and construction organizations generate diverse data sources, while supervising and owner organizations analyze the data to assess and mitigate risks. The proposed data bank aims to connect these four organizations into a consortium. It undergoes organizational authentication, provides distributed database interaction interfaces, and ensures data consistency, tamper resistance, and modification traceability. Additionally, the data bank offers data maintenance and equitable sharing services for the consortium.

The data bank leverages the anti-tampering and traceability features of blockchain to manage the distributed storage of data generated from tunnel SHM and construction activities. Diverse and heterogeneous data will be collected and transmitted to the blockchain-distributed storage interface through the data bank network. The data bank then integrates, calculates, records, and stores this data based on a designated smart contract. Organizations within the consortium can maintain and query the data in the data bank with authorized certification at any time, as well as perform higher-level tasks such as risk management and project management. The architecture of the proposed data bank is illustrated in Fig. 3a, a and the organizations involved in the consortium are shown in Fig. 3b.

Fig. 3b depicts the other elements of the data bank. Core miners are the most crucial of these elements. They serve as key components for maintaining blockchain operations. Similar to a bank's financial management system, core miners function as a management system within a data bank. They are responsible for implementing the consensus algorithm, managing request orders, endorsing organizations, and maintaining distributed data consistency. Each organization requires at least one peer node (called an anchor peer node, an isolated running machine, or an environment, such as a cloud server or Docker container) to communicate and interact with core miners. The interaction logic follows the smart contract deployed on each peer node. The numbers of core miners, organizations in the consortium, and peer nodes in each organization were adjustable. In the proposed data bank, the number of organizations is set to 4, and the number of core miners and peer nodes can be determined based on the actual requirements.

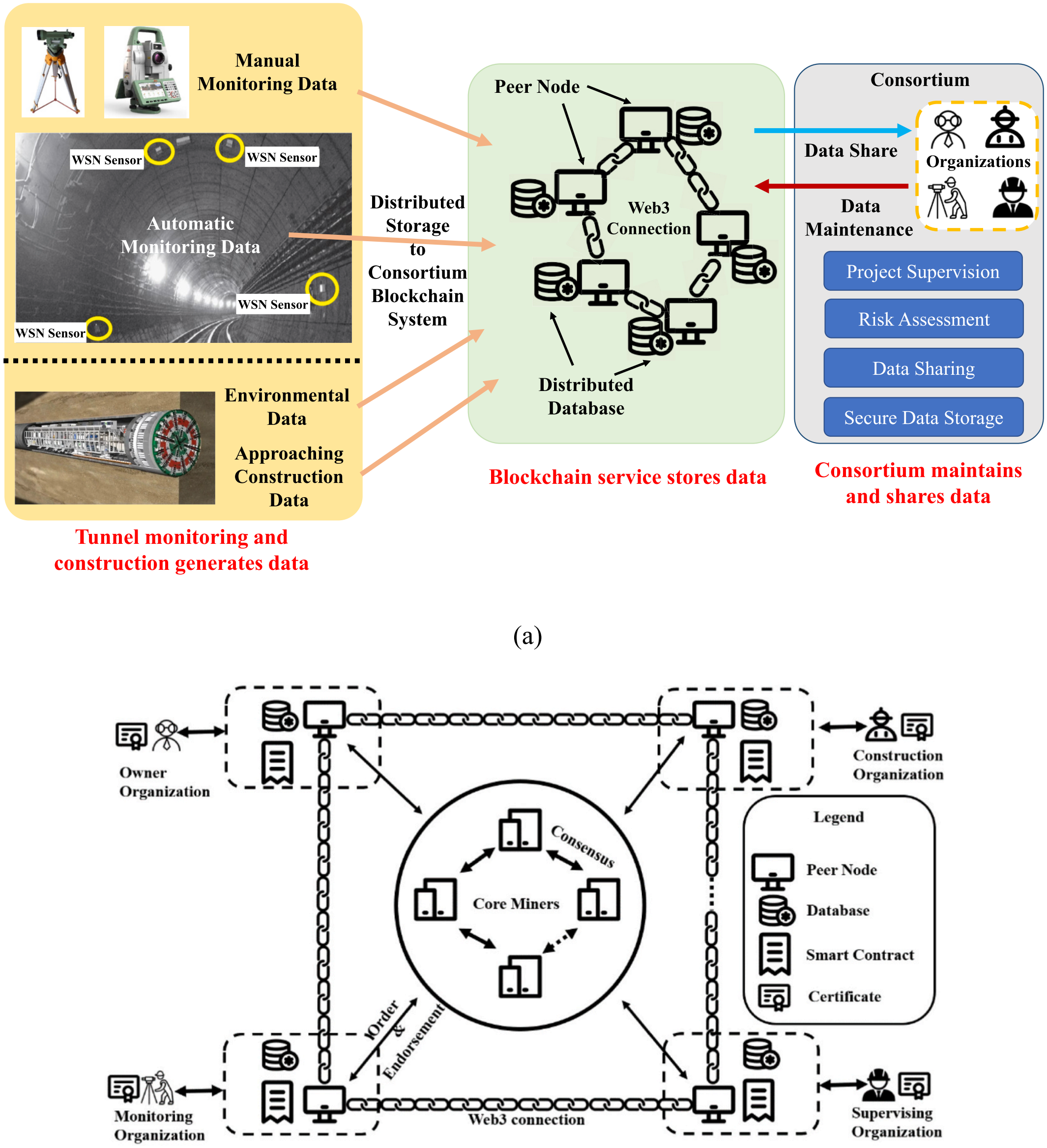
3.2. Smart contract of the traceable data bank

In the designed tunnel data bank, each organization is required to maintain an identical smart contract. If there is any change in the data format, the smart contract is updated and the old one is overwritten with a higher version number. This design pattern differs from that of blockchain smart contracts, which use commercial logic. The benefits of this design pattern are twofold: it is sufficiently concise and achieves the sharing of multi-source and heterogeneous data. The smart contract includes a unified data structure definition, operation logic for the addition, modification, and query of data in the data bank, and early warning calculation. A detailed definition is provided below: In addition, the application section describes a specified smart contract for detailed implementation.

3.2.1. Unified data structure

The unified data structure contains all the necessary information for a tunnel SHM project, which should be discussed and agreed upon by all organizations in the consortium. The specific process of smart contract creation and deployment on the blockchain is illustrated in Fig. 4.

Regarding the smart contract illustrated in Fig. 4, it is important to note that the tunnel data bank is data-driven, with tunnel SHM data being the most crucial information stored within it. This data can originate from monitoring organizations, such as pictorial information, WSN sensors, and manual reports, as well as from construction organizations, including construction progress and machine status). Additionally, with the widespread use of cameras and point cloud scanners, tunnel SHM data now includes not only numerical data but also large-sized files such as images and point clouds. For smaller, structured data, relevant fields can be stored directly in smart contracts. However, for larger data types like images, videos, or point clouds, storing and transmitting them directly in the data bank could place significant bandwidth pressure on



(a)

(b)

Fig. 3. Framework for the proposed data bank. (a) Framework for the interaction and function of the data bank; (b) Framework for the consortium blockchain of the data bank.

the consortium blockchain system. An alternative approach is to store large data on a public cloud server (e.g., Alibaba Cloud OSS), compute the hash value of the content using a digest algorithm (e.g., SHA256), and then upload the storage path along with the hash value to the data bank. The following section of the paper shows a designed heterogeneous tunnel SHM data structure with WSN monitoring, nearby construction progress and machine status as an example.

3.2.2. Interaction logic with tunnel data bank

The interaction between organizations and tunnel data bank revolves around data management, traceability, and sharing. To address the challenge, this paper proposes strict adherence to these three principles.

1) Organization authorization principle: Only the creator of the data has the right to modify the corresponding entry.
2) Modification tracking principle: Operations such as adding or modifying data in the data bank are recorded.
3) Query-free principle: All organizations within the same consortium have access to any data and modification records.

A data bank's distributed database typically has a key-value database structure. In such a key-value database, the key is unique and identifies the stored data entry. Therefore, stringent key specifications are required to prevent conflicts among diverse data sources. In the proposed data bank, strict segregation is maintained for data managed by monitoring and construction organizations. For data managed by monitoring organizations, identification is based on the device number and monitoring time. For data managed by construction organizations, identification is based on data time. When an organization adds or modifies data to the data bank, the smart contract first checks whether the organization has the corresponding permissions. Subsequently, the number and format of the parameters provided by the organization are verified. Only after passing the above two checks can the subsequent operations be executed. This double-check mechanism ensures secure distributed storage. Moreover, if a data entry is modified, the

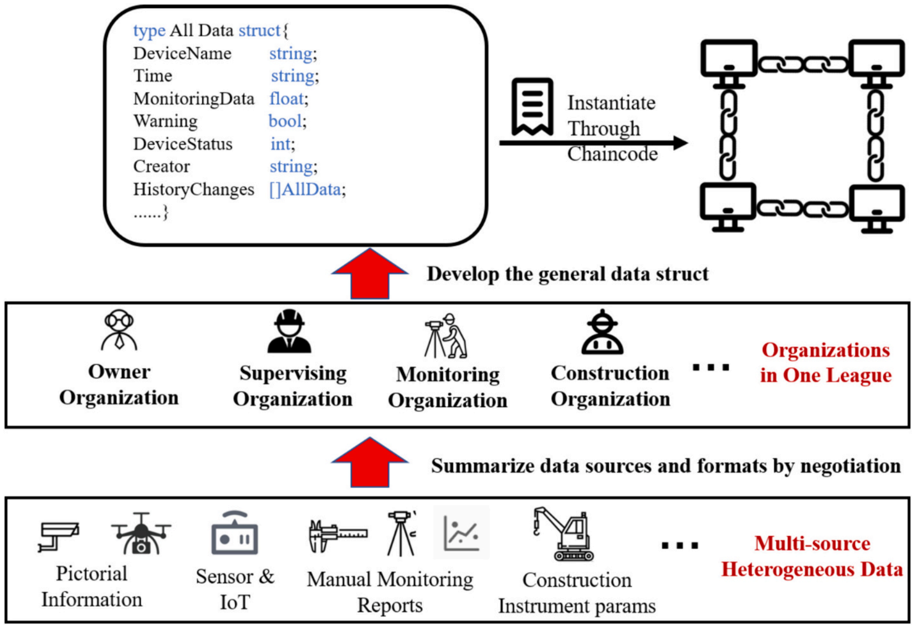


Fig. 4. Process of establishment of unique and shared smart contract.

modification time and previous state are recorded. Public records ensure the traceability of any modifications. In contrast, querying data and modifying records are convenient and straightforward. Not all organizations within the consortium need to “pay for” the data bank to access the data, making it highly conducive to data sharing.

In the various stages of data management, the most crucial operations are creation and modification. The main difference between these two operations is whether the key already exists in the database. Before an organization operates on a specific entry, a check for the existence of a key is conducted. If a key did not exist, it was created. Otherwise, it is modified, as illustrated by the orange background box in Fig. 5. The entry states before and after the modification operation are recorded in the generated transaction.

3.2.3. Calculation and warning of monitoring increments

The monitoring increment refers to the difference between the value of the physical quantity monitored at the current moment and its historical state. To calculate the increment, both current and historical monitoring values were obtained. If the historical data are empty (i.e., the current data are the first entries), the current data are initialized without an increment calculation. Otherwise, the historical monitoring data are queried first and the incremental value at the current moment is calculated based on the queried historical value. The incremental value is recorded in the corresponding field of entry and stored in the distributed key-value databases of each organization. Furthermore, a threshold for increments was designed for smart contracts. If the increment exceeds the threshold, the monitoring alert state is activated. The activated alert state is stored in the database and simultaneously sent to the supervising and owner organizations, reminding them to take timely action. The entire process is depicted by the green background box in Fig. 5.

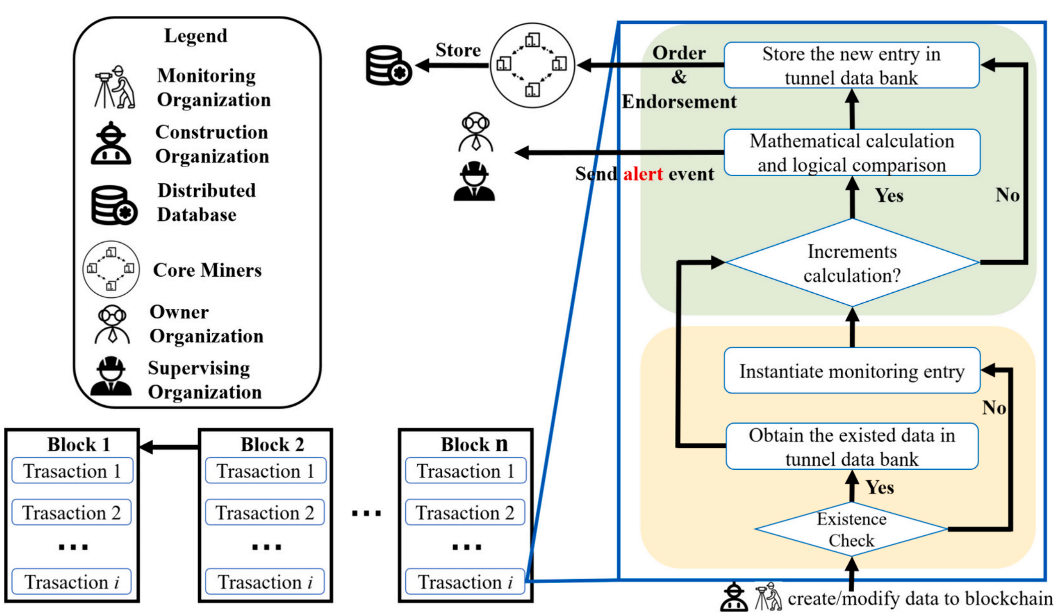


Fig. 5. Interaction logic of entry creation and modification in data bank.

4. Comprehensive data flow paths

The core components of the tunnel data bank, including the consortium framework and smart contracts, were introduced in the previous sections. Additionally, it is crucial to design external network applications for the consortium blockchain, specifically to create pathways for uploading diverse data from multiple sources to a tunnel data bank. The purpose of the design was to enhance the functionality of the data bank and improve user experience, akin to the development of front-end and back-end components in software engineering. Various interfaces for uploading multisource heterogeneous data are provided to achieve the desired functionality of the data bank, supporting manual input, automated data transfer, and information extraction modules based on artificial intelligence. The comprehensive data-flow paths are depicted in Fig. 6, with two paths designed to monitor the data collected by automated devices and one path designed for manual input.

The first two paths are designed for monitoring data from automated devices. Data from both paths are initiated at local gateways, backed up on public cloud servers, and then uploaded to the distributed databases of the data bank. The key difference between these paths is that the first path includes an information-extraction module, which is optional, but can significantly enhance the process by discovering patterns in large data sets, contributing to information extraction, data refinement, and simplified storage. The third path, for manually uploaded data, is much simpler and requires only a manual input interface for data transfer. Additionally, concise interfaces are provided for querying the data in the data bank and receiving alert event messages.

5. Application and test of the tunnel data bank

In this section, a specific tunnel data bank was created and tested. The framework and smart contracts were developed based on the guidelines described in Section 3. The data structure and interaction logic of smart contracts are customized to meet specific requirements, ensuring efficient data management and seamless interaction. The data bank serves the monitoring project of the Hangzhou Metro Line 4, Zhejiang Province, China. The framework was developed using Hyperledger Fabric technology, leveraging its robust features for secure and scalable blockchain solutions. The subsequent sections provide comprehensive details on various aspects including the monitoring project background, consortium blockchain composition, smart contracts, data flow paths, and performance test results.

5.1. Background of the monitoring project

The construction of a shield tunnel passing underneath existing tunnels can significantly interfere with the latter, leading to deformation. In this monitoring project, the Hangzhou Metro Line 4 was impacted by underlying shield tunneling. Two shield tunnels, known as the left and right lines, were being constructed sequentially beneath the existing Line 4 tunnels. These existing tunnels, located in muddy silty clay containing mica fragments, are constructed as a single circular structure made from staggered steel-reinforced concrete segments. The tunnels have an outer diameter of 6.2 m, a segment thickness of 0.35 m, and a ring width of 1.2 m. They were not reinforced in advance, making them highly susceptible to significant differential settlement during the construction of the shield tunnels. The right line of the existing tunnels is closer to the new shield tunnels, with a minimum distance of only 4.58 m. Therefore, the right line is the primary focus of the monitoring project, where monitoring devices will be installed. The new shield tunnels, located in muddy silty clay and medium sand, have an inner diameter of 6.1 m, an outer diameter of 6.9 m, with a segment thickness of 400 mm, a ring width of 1.5 m, and a wedge-shaped segment of 48.5 mm. The sectional diagram is shown in Fig. 7a.

The planar relationship between the monitored existing tunnel and the underlying shield tunnels is illustrated in Fig. 7b. The new tunnels intersect the existing tunnel at an angle of 65°. Before the start of the monitoring project, the left line of the new tunnel had already passed through the existing tunnel, while the right line was still forthcoming.

In this monitoring project, two types of WSN devices were utilized: a Vertical Displacement Monitoring System (VDMS) for tunnel longitudinal settlement monitoring and a Wireless Laser Rangefinder Sensor (WLRS) for tunnel lateral convergence monitoring. The physical appearances of these devices are shown in Fig. 8.

The VDMS longitudinal settlement calculation method is as follows: The VDMS consists of several interconnected measurement units (called cells) and is fixed to the lining wall. Assuming that the inclination and the cell length of the i^{th} cell of the VDMS are known, φ_i and L_i, the settlement of the monitoring section can be measured and calculated sequentially cell by cell. The settlement curve was then obtained, as shown in Fig. 9a. Assuming that the settlement at the start is known, for example, y_0, the settlement y_p at point x_p is equal to Eq. 1:

$$y_p = y_0 + \sum_{i=1}^{k-1} (L_i sin\varphi_i) + \left(x_p - \sum_{i=1}^{k-1} L_i \right) sin\varphi_k \tag{1}$$

where k indicates that point x_p is located within the k^{th} cell and L_i is the

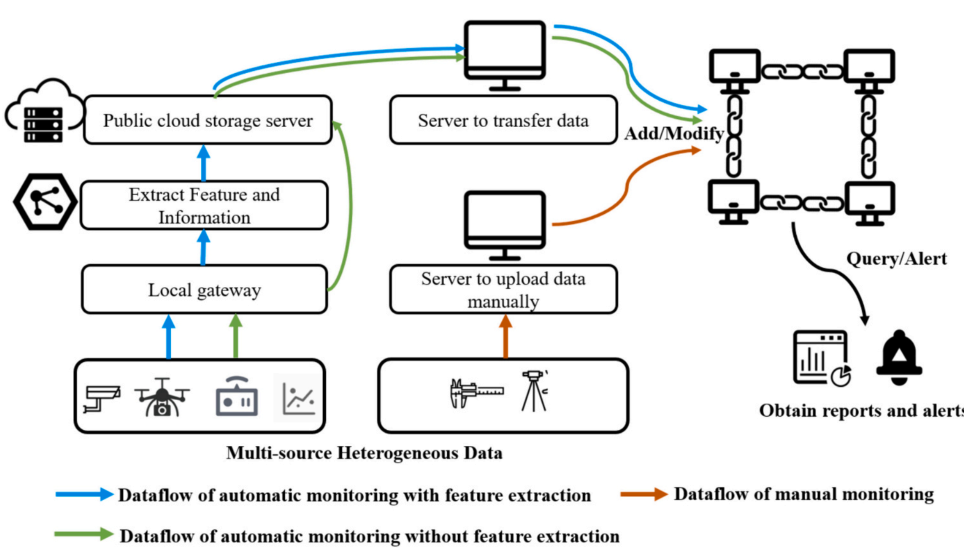


Fig. 6. Dataflows of proposed data bank.

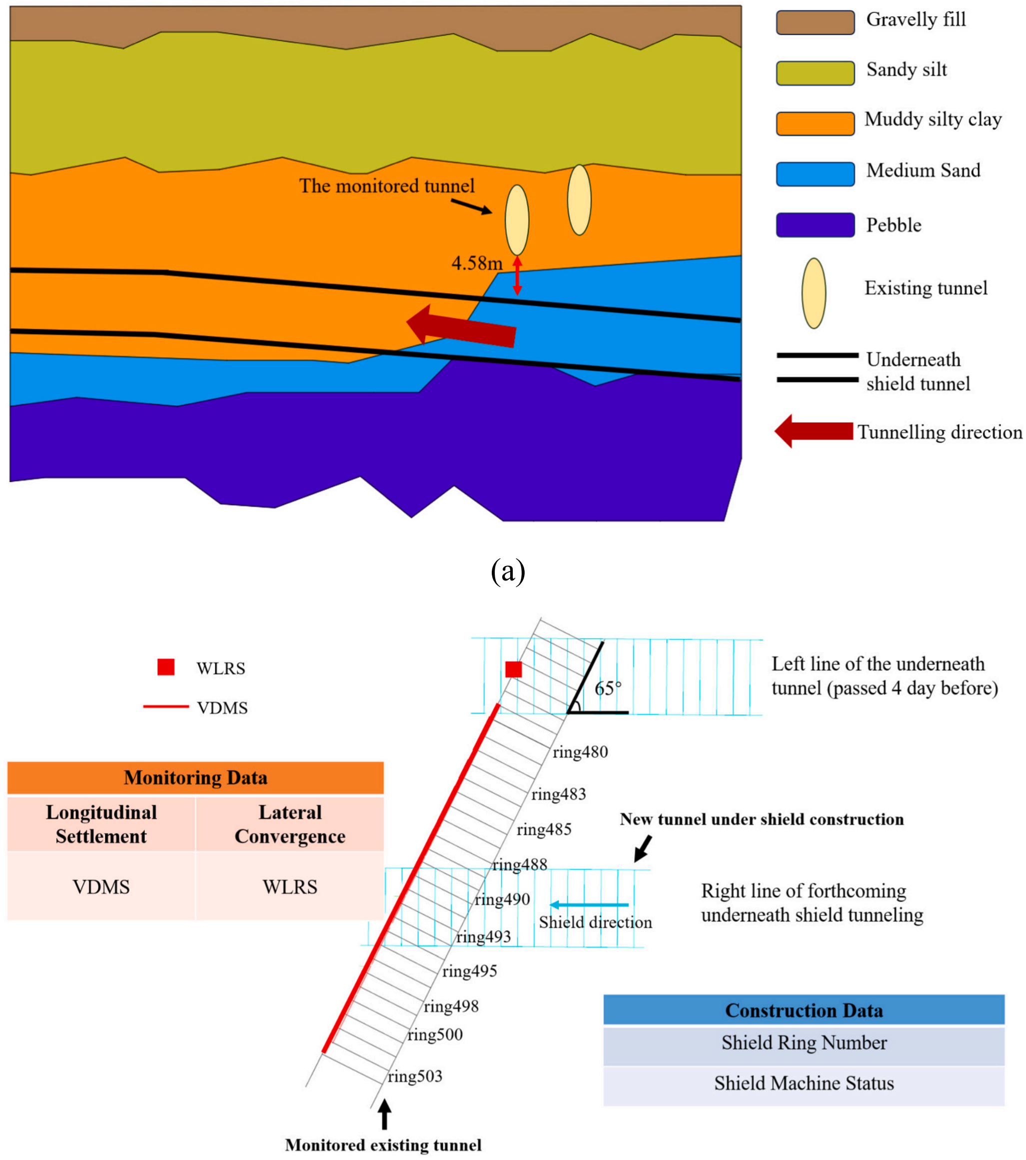


Fig. 7. Spatial relationship between tunnels and overview of sensor deployment. (a) Soil layer distribution and spatial relationship of tunnels; (b) Planar relationship of tunnels and deployment of monitoring instruments.

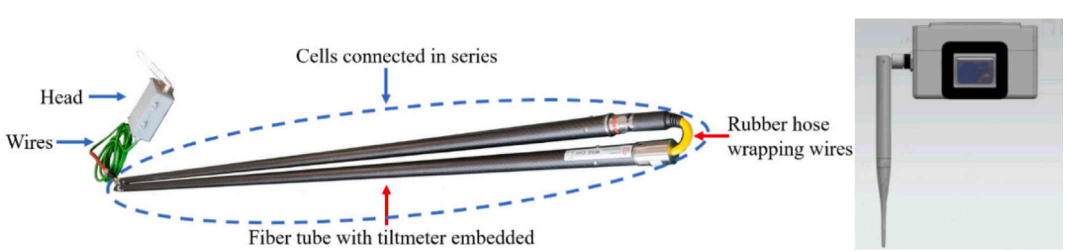


Fig. 8. Monitoring devices. (a) VDMS; (b) WLRS.

length of the VDMS cell [48].

The convergence calculation method for the WLRS is as follows: The WLRS was installed at the waist of the ring, and the laser was emitted horizontally towards the opposite side of the ring to measure the lateral convergence of the tunnel, as shown in Fig. 9b. Assuming that the initial tunnel diameter is d_0 and the i^{th} measurement is d_i, the convergence

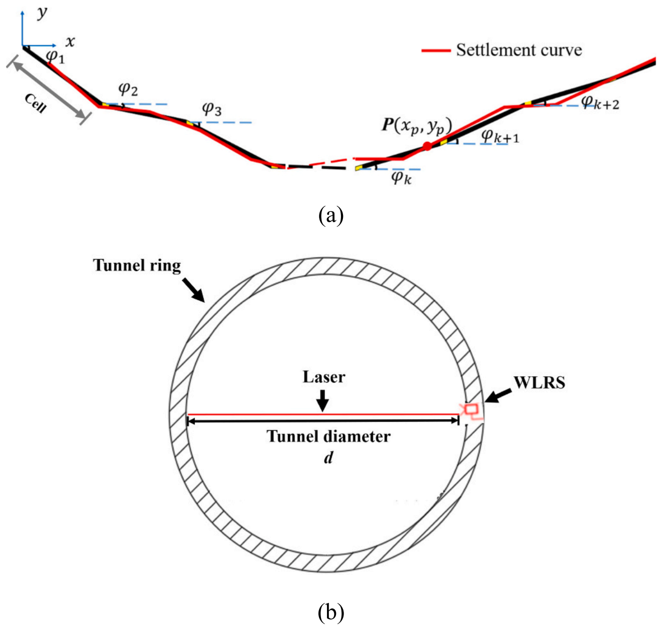


Fig. 9. Tunnel deformation calculation method for WSN devices. (a) Longitudinal settlement calculation for VDMS; (b) Lateral convergence calculation for WLRS.

deformation c_i is given by Eq. 2:

$$c_i = d_i - d_0 \tag{2}$$

Both instruments are connected to smart gateways that upload the monitoring data to public cloud servers for storage. The VDMS and WLRS are installed at locations where significant deformation of the existing tunnel is most likely to occur. The VDMS was installed between rings 480 and 503 of the existing monitored tunnel, spanning the construction area of the future shield tunnel. The WLRS was installed at ring 476 of the monitored existing tunnel, directly above the already passed shield tunnel construction area. The installation scenes for both are shown in Fig. 10, and the planar layouts are shown in Fig. 7b.

In addition to monitoring the structural deformation of the existing tunnels, it is essential to also monitor and control the construction of the new shield tunnel. The construction organization will provide updates on the progress of the shield tunnel construction, including the ring number reached on the current date and the status of the tunnel-boring machine equipment, as detailed in the table shown in Fig. 7b. In summary, the data managed by the tunnel data bank is diverse and originates from multiple sources, including WSN monitoring data from the monitoring organization and construction information from the construction organization.

5.2. Implementation of the tunnel data bank

In this section, the framework of the consortium blockchain is first established and then the details of the smart contract are presented.

5.2.1. Establishment of the consortium blockchain

To ensure construction safety, the monitoring organization conducts structural deformation monitoring in key sections of the existing tunnel. The construction organization is responsible for managing construction progress. Additionally, the supervising organization and owner organization are responsible for ensuring overall construction safety. Therefore, a consortium consisting of these four organizations was established.

The consortium blockchain application was deployed using Hyperledger Fabric v2.5 [49]. The runtime environments are listed in Table. 1. For ease of research, the entire consortium blockchain was deployed and operated within a local area network.

The orderer and peer nodes of the organizations are both launched as Docker containers in the background of the operating system, as shown in Fig. 11. Mapping from the Docker containers to the peer nodes is presented in Table. 2. The number of orderers was one. Owing to the negligible network latency within the local area network, the probability of an order crashing is very low. Therefore, setting the number to 1 satisfies these requirements. However, the orderer still follows the raft consensus algorithm, while the individual orderer is always responsible for ordering and confirming.

5.2.2. Data structure in smart contract

The data structure designed for unique and shared smart contracts is shown in Fig. 12. It consists of the following components.

(1) *ConstructorData*: This includes the construction progress and machine status of the new shield tunnel.
(2) *MonitorData*: This contains the WSN monitoring data on longitudinal settlement and lateral convergence.

Table 1
Simulation environment configuration.

Config Item	Config Info
Operation System	Ubuntu 20.04
CPU	AMD RYZEN 5000
Memory	16GB
Storage	SSD 256G

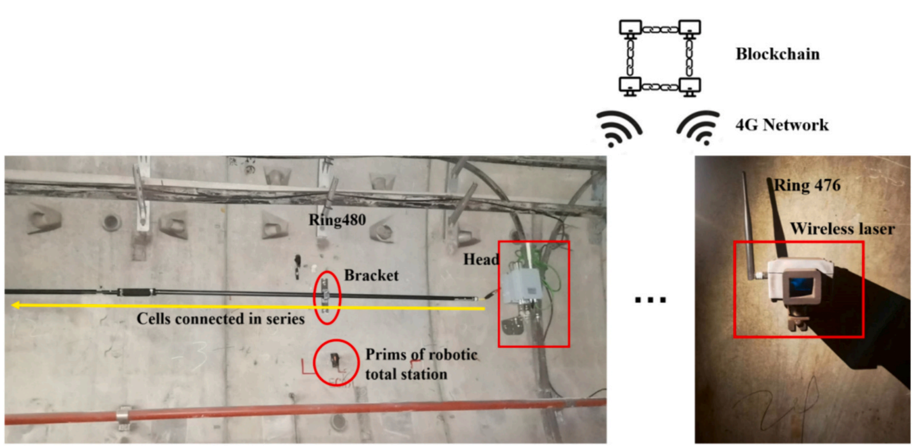


Fig. 10. Photo of VDMS and WLRS on site.

```
CONTAINER ID   IMAGE                               COMMAND                  CREATED         STATUS                   NAMES
9c780c7e089f   hyperledger/fabric-tools:latest     "/bin/bash"              6 seconds ago   Up Less than a second cli
49cc1549f85e   hyperledger/fabric-peer:latest      "peer node start"        6 seconds ago   Up 1 second              peer0.org4.example.com
1bdc31c374e3   hyperledger/fabric-peer:latest      "peer node start"        6 seconds ago   Up Less than a second    peer0.org1.example.com
78597b54fd28   hyperledger/fabric-peer:latest      "peer node start"        6 seconds ago   Up 1 second              peer0.org2.example.com
d0213974ffb6   hyperledger/fabric-peer:latest      "peer node start"        6 seconds ago   Up 1 second              peer0.org3.example.com
77480ee0acaf   hyperledger/fabric-orderer:latest   "orderer"                6 seconds ago   Up 3 seconds             orderer.example.com
8a82886ea677   couchdb:3.3.2                       "tini -- /docker-ent…"   6 seconds ago   Up 3 seconds             couchdb2
37e83906b9c8   couchdb:3.3.2                       "tini -- /docker-ent…"   6 seconds ago   Up 2 seconds             couchdb1
a47e97caa8b1   couchdb:3.3.2                       "tini -- /docker-ent…"   6 seconds ago   Up 2 seconds             couchdb3
8c4c308b2b49   couchdb:3.3.2                       "tini -- /docker-ent…"   6 seconds ago   Up 3 seconds             couchdb4
roach@roach-virtual-machine:~/go/src/github.com/latest-test/fabric-samples/test-network$
```

Fig. 11. Docker containers representing components of consortium blockchain.

Table 2
Mapping from Docker containers to peer nodes.

Docker Containers	Peer Nodes
cli	Client node
peer0.org1.example.com	Anchor peer node of monitoring organization
peer0.org2.example.com	Anchor peer node of construction organization
peer0.org3.example.com	Anchor peer node of supervising organization
peer0.org4.example.com	Anchor peer node of owner organization
orderer.example.com	orderer
couchdb1	Key-value pair database of monitoring organization
couchdb2	Key-value pair database of construction organization
couchdb3	Key-value pair database of supervising organization
couchdb4	Key-value pair database of owner organization

(3) *Data*: This contains a series of *ConstructorData* and *MonitorData*, and is directly used when data entries are queried by date.
(4) *History*: This is used when the modified records are queried.

In the distributed key-value database, *MonitorData* and *ConstructorData* are stored with unique keys for distinction and identification. For *MonitorData*, the key is the monitoring time and the WSN device number, whereas for *ConstructorData*, the key is formed based on the date. Tables 3 and 4 provide examples to illustrate the field values for *MonitorData* and *ConstructorData*. In the context of *DeviceStatus* and *NeedAlert*, the default value is zero for normal status and one for alert status.

5.2.3. Permission policy in smart contract

Strict permission policies have been established for smart contracts. Before any operation changes the state of the database, the organization's permission is checked. This was implemented by calling the *GetMSPID*() function to obtain the membership identification (MSPID) for each organization. The MSPIDs of the four organizations are listed in Table. 5.

5.2.4. Interaction logic in smart contract

Seven functions are defined in smart contracts for the interaction between organizations and the tunnel data bank, as listed in Table. 6. Specifically, only the construction organization is given modification permission because the data are manually input and prone to errors. The calculation of increments for the monitoring data was included in *AddDataByKeyforMonitor*(). The settlement delta threshold S_T and

```go
type ConstructorData struct {
    Time          string `json:"time"`
    Process       string `json:"process"`
    MachineStatus int    `json:"machinestatus"`
    NeedAlert     int    `json:"needalert"`
}

type MonitorData struct {
    Time         string  `json:"time"`
    MinitorType  string  `json:"monitortype"`
    DeviceNo     string  `json:"deviceno"`
    OrigData     float32 `json:"origdata"`
    CalcData     float32 `json:"calcdata"`
    BaseTime     string  `json:"basetime"`
    Delta        float32 `json:"delta"`
    DeviceStatus int     `json:"devicestatus"`
    NeedAlert    int     `json:"needalert"`
}

type History struct {
    TxId      string    `json:"txid"`
    Timestamp time.Time `json:"timestamp"`
    Data      *Data     `json:"data"`
}

type Data struct {
    Time             string             `json:"time"`
    MonitorDatas     []*MonitorData     `json:"monitordatas"`
    ConstructorDatas []*ConstructorData `json:"constructordatas"`
    NeedAlert        int                `json:"needalert"`
}
```

Fig. 12. Data structure designed in smart contract.

Table 3
Example of *MonitorData*.

MonitorData Fields	Example
Time	20,220,201,040,500
MonitorType	Settlement
DeviceNo	SN1750315
OrigData	87°
CalcData	1.74
BaseTime	20,220,201,030,500
Delta	0.03
DeviceStatus	0
NeedAlert	0

Table 4
Example of *ConstructorData*.

ConstructorData Fields	Example
Time	20,220,201,000,000
Process	Ring480
MachineStatus	0
NeedAlert	0

Table 5
MSPIDs of four organizations.

Organization Name	MSPID Name
Monitoring Organization	Org1MSP
Construction Organization	Org2MSP
Supervising Organization	Org3MSP
Owner Organization	Org4MSP

Table 6
Functions defined in smart contract.

Function Name	Applicable Scenarios
AddDataByKeyforMonitor()	Monitoring organization creates data
AddDataByKeyforConstructor()	Construction organization creates data
UpdateDataforConstructor()	Construction organization modifies existing data
QueryDataByDate()	Any organization queries data by date
QueryDataByKeyforMonitorData()	Any organization queries monitoring data by key
QueryDataByKeyforConstructorData ()	Any organization queries construction data by key
QueryHistoryByDate()	Any organization queries modification records by date

convergence delta threshold C_T variables were set globally. When the i^{th} monitoring data entry is created, settlement s_i is calculated using Eq. 1, and convergence c_i was calculated according to Eq. 2. Then the *Delta* for settlement Δs_i is calculated by Eq. 3:

$$\Delta s_i = s_i - s_{i-1} \tag{3}$$

And the *Delta* for convergence Δc_i is calculated by Eq. 4:

$$\Delta c_i = c_i - c_{i-1} \tag{4}$$

If the $MonitorType_i$ is equal to “Settlement”, the $NeedAlert_i$ is calculated by Eq. 5:

$$NeedAlert_i = NeedAlert_{i-1} \vee (\Delta s_i > S_T) \tag{5}$$

If the $MonitorType_i$ is equal to “Convergence”, the $NeedAlert_i$ is calculated by Eq. 6:

$$NeedAlert_i = NeedAlert_{i-1} \vee (\Delta c_i > C_T) \tag{6}$$

Where the “$\vee$” is logical OR operator.

For $NeedAlert_i$ in *ConstructorData*, only the *MachineStatus* is checked.

5.3. Dataflow of the tunnel data bank

The WSN monitoring dataflow corresponds to the second path type described in Section 4. The construction dataflow was of the third type. Construction organizations must upload data manually.

5.4. Performance testing

5.4.1. Traceability of the tunnel data bank

Only the construction organization was allowed to modify the construction data to correct potential occasional manual errors in the tunnel data bank. However, any organization can call *QueryHistoryByDate*() to track change records. For example, a construction data entry stated that the shield was at ring 480 on February 6, 2022. The construction organization then calls *UpdateDataforConstructor*(), which changes the progress to Ring 490. After calling *QueryHistoryByDate*(), the tunnel data bank traced the changes, as shown in Fig. 13. The results are summarized in Table. 7.

5.4.2. Delay of the tunnel data bank

Delay is defined as the time required for a user to send and complete a request. To comprehensively test the delay in the tunnel data bank, all four organizations interacted synchronously with it. The monitoring organization calls *AddDataByKeyforMonitor*() to upload the WSN monitoring data and the construction organization calls *AddDataByKeyforConstructor*() to upload the construction progress data. All four organizations call *QueryDataByKeyforMonitorData*() and *QueryDataByKeyforConstructorData*(). The call loops are executed using different processes.

In this regard, all monitoring data for Hangzhou Metro Line 4 (including hourly VDMS and WLRS monitoring) and all construction data (daily reported excavation progress reports) will be transmitted to the chain at short notice. The monitoring period was from January 1, 2022, to August 1, 2022. Each WSN device had approximately 5760 measurement entries, resulting in approximately 86,400 records of monitoring data (four VDMS instruments and one WLRS). In addition, approximately 240 construction data records represent different construction processes.

Additionally, for the Hyperledger Fabric blockchain service, *MaxMessageCount*, *BatchTimeout*, and *PreferredMaxBytes* are three important parameters that significantly affect the latency and service stability [49]. Different combinations of values for these three parameters are used to construct the consortium blockchain separately. An optimal parameter combination that minimizes the average delay is sought. The average delay is calculated using Eq. 7:

$$t_{avg} = \frac{1}{O}\frac{1}{N}\sum_{i=1}^{O}\sum_{j=1}^{N} t_{i,j} \tag{7}$$

where t_{avg} is the average delay of all interactions of all organizations. O is the number of organizations (four), and N represents the number of interactions of each organization (six).

The combinations of the three parameters and the average delay results are listed in Table. 8. Furthermore, the maximum delay of each interaction is not more than 0.8 s. Considering that the upload interval for the actual project is 1 h, the delay generated by sorting, storing, and sending fully meets the requirements. This demonstrated the applicability of the database to real monitoring projects.

5.4.3. Comparison with conventional SHM system

Blockchain-based SHM systems are relatively rare. However, compared to common centralized and distributed storage systems, our consortium blockchain-based tunnel data bank offers several advantages, as shown in Table. 9.

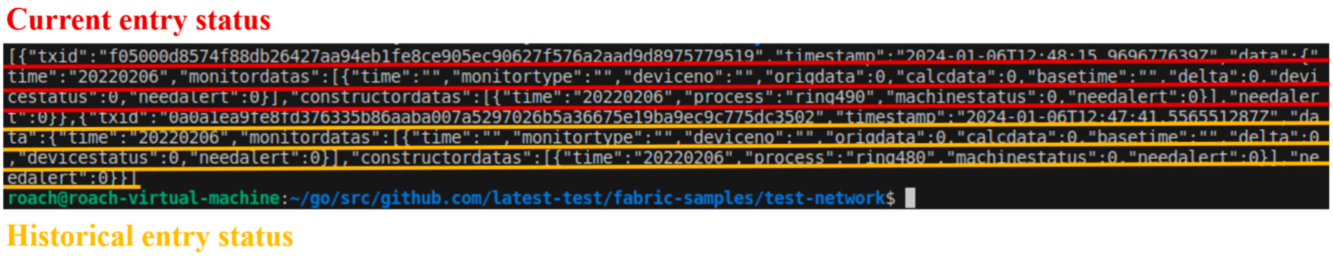


Fig. 13. Tracing of construction data modifications.

Table 7
Organized tracing records of results shown in Fig. 13.

Fields	Historical Status	Current Status
Transaction ID	0a0a1ea9fe8fd376335b86aaba007a5297026b5a36675e19ba9ec9c775dc3502	f05000d8574f88db26427aa94eb1fe8ce905ec90627f576a2aad9d8975779519
Status change time	2024–01-06 T12:47:41.556551287	2024–01-06 T12:48:15.969677639
Monitoring data	/	/
Construction time	20,220,206,000,000	20,220,206,000,000
Construction progress	ring480	ring490
Machine status	0	0

Table 8
t_{avg} of all combinations of three crucial parameters.

Combination	*MaxMessageCount*	*BatchTimeout*	*PreferredMaxBytes*	t_{avg}
No.1	70	2 s	2 MB	0.439 s
No.2	70	2 s	8 MB	0.424 s
No.3	**70**	**4 s**	**2 MB**	**0.373 s**
No.4	70	4 s	8 MB	0.435 s
No.5	90	2 s	2 MB	0.445 s
No.6	90	2 s	8 MB	0.416 s
No.7	90	4 s	2 MB	0.384 s
No.8	90	4 s	8 MB	0.392 s

6. Conclusions

To address data isolation in tunnel SHM, a traceable tunnel data bank was innovatively proposed for SHM data management, sharing, and traceability. The applicability of the data bank is outlined as follows.

(1) Framework: The tunnel data bank includes a consortium blockchain that connects the supervising, construction, owner, and monitoring organizations. This setup integrates previously isolated units, enabling peer-to-peer communication for distributed data storage and sharing.
(2) Smart contracts: A unified data structure for multi-source heterogeneous data facilitates collaborative maintenance, querying, and modifications. It records the history of data modification for authentication and accountability, which is highly beneficial for data authentication and incident accountability. It also provides functions for calculating monitoring increments and issuing alerts based on historical data.
(3) Data flow patterns: The platform supports manual input and automatic uploads that are compatible with most current tunnel SHM equipment.
(4) Implementation: Built on Hyperledger Fabric: The platform was tested with a WSN and construction data from Hangzhou. The performance tests showed low latency and high stability, ensuring timely and secure data upload, storage, and access.

Table 9
Comparison of consortium-blockchain-based tunnel data bank with mainstream centralized and distributed storage systems.

Feature	Blockchain Storage	Distributed Storage	Centralized Storage
Data Integrity	**High**	Medium	Low
Transparency	**High**	Medium	Low
Decentralization	**High**	Medium	Low
Scalability	Medium	**High**	Medium
Performance	Medium	**High**	**High**
Fault Tolerance	**High**	Medium	Low
Security	**High**	Medium	Low

Despite these advancements, there are some limitations to consider:

(1) Alert judgment: A smart contract relies on a predetermined threshold for alert judgment. If the threshold changes, then the smart contract must be updated.
(2) Delay for extremely high-frequency interactions: The data bank may not meet the requirements for interactions with extremely high frequencies such as those occurring in less than 1 s. The integration of cache or queue mechanisms can address this problem.

Future work could focus on developing modular and automated smart contract designs to enhance the user-friendliness of writing smart contracts through low-code approaches. Additionally, optimizing user interface logic and improving the user experience and management convenience of the platform should be pursued. Exploring the integration of artificial intelligence modules could also enable advanced data analysis, anomaly detection, and predictive maintenance in tunnel SHM.

CRediT authorship contribution statement

Dong-Ming Zhang: Resources, Funding acquisition, Conceptualization. **Cong Nie:** Writing – original draft, Methodology, Data curation.

Jin-Zhang Zhang: Writing – review & editing, Validation, Supervision. **Hong-Wei Huang:** Supervision, Investigation, Conceptualization. **Xu Huang:** Project administration, Investigation.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work is substantially supported by the National Key Research and Development Program of China (2021YFF0502200), Key Technologies of Intelligent Monitoring System for Shield Tunneling through Soft Soil Layer in Close Proximity to Operational Tunnels (2021-KY-04), Academician and Expert Workstation of Yunnan Province (Grants No. 202205AF150015), and the Science and Technology Innovation Project of YCIC Group Co., Ltd. (Grants No. YCIC-YF-2022-15). The financial support is gratefully acknowledged.

References

[1] J.Y. Yan, R.L. Wang, Cause analysis and typical characteristics of settlement and transverse convergence deformation of metro tunnel in soft ground in Shanghai, J. Nat. Disasters 27 (4) (2018) 178–187, https://doi.org/10.13577/j.jnd.2018.0424.
[2] S.L. Shen, H.N. Wu, Y.J. Cui, Z.Y. Yin, Long-term settlement behaviour of metro tunnels in the soft deposits of Shanghai, Tunn. Undergr. Space Technol. 40 (2014) 309–323, https://doi.org/10.1016/j.tust.2013.10.013.
[3] China Association of Metros, Urban Rail Transit 2022 Annual Statistical and Analysis Report, China Metros 4, 2023, pp. 13–15, https://doi.org/10.14052/j.cnki.china.metros.2023.04.002.
[4] Y.B. Luo, J.X. Chen, W.Z. Xi, P.Y. Zhao, J.Z. Li, X. Qiao, Q. Liu, Application of a total station with RDM to monitor tunnel displacement, J. Perform. Constr. Facil. 31 (4) (2017) 04017030, https://doi.org/10.1061/(ASCE)CF.1943-5509.0001027.
[5] C.P. Zhang, D.L. Zhang, J.J. Luo, M.S. Wang, J.P. Wu, Remote monitoring system applied to the construction of metro station undercrossing existing metro tunnel, Rock Soil Mech. 30 (6) (2009) 1861–1866, https://doi.org/10.16285/j.rsm.2009.06.013.
[6] J.G. Zhou, H.L. Xiao, W.W. Jiang, W.F. Bai, G.L. Liu, Automatic subway tunnel displacement monitoring using robotic total station, Measurement 151 (2020) 107251, https://doi.org/10.1016/j.measurement.2019.107251.
[7] A.B. Noel, A. Abdaoui, T. Elfouly, M.H. Ahmed, A. Badawy, M.S. Shehata, Structural health monitoring using wireless sensor networks: a comprehensive survey, IEEE Commun. Surv. Tutorials 19 (3) (2017) 1403–1423, https://doi.org/10.1109/COMST.2017.2691551.
[8] G.D. Zhou, T.H. Yi, M.X. Xie, H.N. Li, J.H. Xu, Optimal wireless sensor placement in structural health monitoring emphasizing information effectiveness and network performance, J. Aerosp. Eng. 34 (2) (2021), https://doi.org/10.1061/(ASCE)AS.1943-5525.0001226.
[9] P.J. Bennett, K. Soga, I. Wassell, P. Fidler, K. Abe, Y. Kobayashi, M. Vanicek, Wireless sensor networks for underground railway applications: case studies in Prague and London, Smart Struct. Syst. 6 (5_6) (2010) 619–639, https://doi.org/10.12989/sss.2010.6.5_6.619.
[10] H. Yamashita, N. Handa, Y. Higashiura, M. Abe, Flexure structural scanner of tip scan type for high-speed scanning tunneling microscopy, e-J. Surf. Sci. Nanotechnol. 18 (2020) 146–151, https://doi.org/10.1380/ejssnt.2020.146.
[11] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/en/bitcoin-paper, 2008.
[12] W. Gavin, Ethereum Yellow Paper: A formal specification of Ethereum, a programmable blockchain, 2014. https://ethereum.github.io/yellowpaper/paper.pdf.
[13] X.F. Piao, M.X. Li, F.C. Meng, H.H. Song, Latency Analysis for Raft Consensus on Hyperledger Fabric, 4th International Conference on Blockchain and Trustworthy Systems (BlockSys) 1679, 2022, pp. 165–176, https://doi.org/10.1007/978-981-19-8043-5_12.
[14] B.W. Jo, R.M.A. Khan, Y.S. Lee, Hybrid blockchain and internet-of-things network for underground structure health monitoring, Sensors 18 (2018) 4268, https://doi.org/10.3390/s18124268.
[15] G.T. Maite, C. Ricardo, Blockchain technology and the sustainable development goals, DYNA 97 (6) (2022) 594–598, https://doi.org/10.6036/10549.
[16] R. Yang, R. Wakefiled, S. Lyu, S. Jayasuriya, F. Han, Y. Xun, X. Yang, A. Gayashan, S. Chen, Public and private blockchain in construction business process and information integration, Autom. Constr. 118 (2020) 103276, https://doi.org/10.1016/j.autcon.2020.103276.
[17] D. Lee, S. Lee, M. Neda, L. Victor, Integrated digital twin and blockchain framework to support accountable information sharing in construction projects, Autom. Constr. 127 (2021) 103688, https://doi.org/10.1016/j.autcon.2021.103688.
[18] K. Prabhat, K. Randhir, K. Abhinav, G. Sahil, S. Satinder, Blockchain and deep learning for secure communication in digital twin empowered industrial IoT network, IEEE Trans. Netw. Sci. Eng. 10 (2022), https://doi.org/10.1109/TNSE.2022.3191601, 2802–281.
[19] Z. Wang, T. Wang, H. Hu, J. Gong, X. Ren, Q. Xiao, Blockchain-based framework for improving supply chain traceability and information sharing in precast construction, Autom. Constr. 111 (2020) 103063, https://doi.org/10.1016/j.autcon.2019.103063.
[20] X. Li, W. Lu, F. Xue, L. Wu, R. Zhao, J. Lou, J. Xu, Blockchain-enabled IoT-BIM data bank for supply chain management in modular construction, J. Constr. Eng. Manag. 148 (2022) 04021195, https://doi.org/10.1061/(ASCE)CO.1943-7862.0002229.
[21] F. Elghaish, R. Hosseini, T. Kocaturk, A. Mehrdad, M. Ledari, Digitalized circular construction supply chain: an integrated BIM-blockchain solution, Autom. Constr. 148 (2023), https://doi.org/10.1016/j.autcon.2023.104746.
[22] Roberto Brandín, Sepehr Abrishami, IoT-BIM and blockchain integration for enhanced data traceability in offsite manufacturing, Autom. Constr. 159 (2024), https://doi.org/10.1016/j.autcon.2024.105266.
[23] X. Tao, Y. Liu, P. Wong, K. Chen, M. Das, J. Cheng, Confidentiality-minded framework for blockchain-based BIM design collaboration, Autom. Constr. 136 (2) (2022) 104172, https://doi.org/10.1016/j.autcon.2022.104172.
[24] X. Tao, M. Das, Y. Liu, J. Cheng, Distributed common data environment using blockchain and interplanetary file system for secure BIM-based collaborative design, Autom. Constr. 130 (7) (2021) 103851, https://doi.org/10.1016/j.autcon.2021.103851.
[25] B.W. Jo, R.M.A. Khan, Y.S. Lee, Hybrid blockchain and internet-of-things network for underground structure health monitoring, Sensors 18 (2018) 4268, https://doi.org/10.3390/s18124268.
[26] J. Xu, H.R. Liu, Q.H. Han, Blockchain technology and smart contract for civil structural health monitoring system, Comput. Aid. Civ. Infrastruct. Eng. 36 (10) (2023) 1288–1305, https://doi.org/10.1111/mice.12666.
[27] M. Gordan, P. Siow, A. Deifalla, O. Chao, Z. Ismail, K. Yee, Implementation of a secure storage using blockchain for PCA-FRF sensor data of plate-like structures, IEEE Access 10 (2022) 84837–84852, https://doi.org/10.1109/ACCESS.2022.3197776.
[28] M. Sidorov, J. Khor, P. Nhut, Y. Matsumoto, R. Ohmura, A public blockchain-enabled wireless LoRa sensor node for easy continuous unattended health monitoring of bolted joints: implementation and evaluation, IEEE Sensors J. 20 (2020) 13057–13065, https://doi.org/10.1109/JSEN.2020.3001870.
[29] F. Franchi, V. Gattulli, F. Graziosi, F. Potenza, Structural health monitoring systems operating in a 5G-based network, Eur. Workshop Struct. Health Monitor. 253 (2023) 89–97, https://doi.org/10.1007/978-3-031-07254-3_10.
[30] Z. Nie, Y. Long, S. Zhang, Y. Lu, A controllable privacy data transmission mechanism for internet of things system based on blockchain, Int. J. Distrib. Sens. Netw. 18 (2022), https://doi.org/10.1177/15501329221088450, 155013292210884.
[31] X.P. Xiong, G.H. Qing, H. Li, Blockchain-based P2P power trading mechanism for PV prosumer, Energy Rep. 8 (2022) 300–310, https://doi.org/10.1016/j.egyr.2022.02.130.
[32] M.R. Perez, B. Gerardo, R. Medina, M. Assoc Comp, Applying modified SHA256 on blockchain using challenge response and off-chain signatures patterns, in: 3rd International Conference on Graphics and Signal Processing (ICGSP), Hong Kong, 2019, pp. 58–61, https://doi.org/10.1145/3338472.3338486.
[33] X.Q. Sun, F.R. Yu, P. Zhang, Z.W. Sun, W.X. Xie, X. Peng, A survey on zero-knowledge proof in blockchain, IEEE Netw. 35 (4) (2021) 198–205, https://doi.org/10.1109/MNET.011.2000473.
[34] A.P. Mohan, R.M. Asfak, A. Gladston, Merkle tree and blockchain-based cloud data auditing, Int. J. Cloud Appl. Comput. 10 (3) (2020) 54–66, https://doi.org/10.4018/IJCAC.2020070103.
[35] Z.H. Xiong, Y. Zhang, D. Niyato, P. Wang, Z. Han, When mobile blockchain meets edge computing, IEEE Commun. Mag. 56 (8) (2018) 33–39, https://doi.org/10.1109/MCOM.2018.1701095.
[36] A.Y. Li, X.H. Wei, Z. He, Robust proof of stake: a new consensus protocol for sustainable blockchain systems, Sustainability 12 (7) (2020), https://doi.org/10.3390/su12072824.
[37] J.W. Yang, J.H. Dai, H.B. Gooi, H.D. Nguyen, A. Paudel, A proof-of-authority blockchain-based distributed control system for islanded microgrids, IEEE Trans. Industr. Inform. 18 (11) (2022) 8287–8297, https://doi.org/10.1109/TII.2022.3142755.
[38] F. Yang, W. Zhou, Q.Q. Wu, R. Long, N.N. Xiong, M.Q. Zhou, Delegated proof of stake with downgrade: a secure and efficient blockchain consensus algorithm with downgrade mechanism, IEEE Access 7 (2019) 118541–118555, https://doi.org/10.1109/ACCESS.2019.2935149.
[39] X.L. Xu, D.W. Zhu, X.X. Yang, S. Wang, L.Y. Qi, W.C. Dou, Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain, ACM Trans. Internet Technol. 21 (1) (2021) 1–17, https://doi.org/10.1145/3395331.
[40] O. Diego, O. John, In search of an understandable consensus algorithm, in: Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference, 2014, pp. 305–320, https://doi.org/10.5555/2643634.2643666.

[41] K.K. Gai, Y.L. Wu, L.H. Zhu, M.K. Qiu, M. Shen, Privacy-preserving energy trading using consortium blockchain in smart grid, IEEE Trans. Industr. Inform. 15 (6) (2019) 3548–3558, https://doi.org/10.1109/TII.2019.2893433.

[42] J.Z. Zhang, H.W. Huang, D.M. Zhang, M.L. Zhou, C. Tang, D.J. Liu, Effect of ground surface surcharge on deformational performance of tunnel in spatially variable soil, Comput. Geotech. 136 (2021) 104229, https://doi.org/10.1016/j.compgeo.2021.104229.

[43] J.Z. Zhang, Z.Q. Liu, D.M. Zhang, H.W. Huang, K.K. Phoon, Y.D. Xue, Improved coupled Markov chain method for simulating geological uncertainty, Eng. Geol. 298 (2022) 106539, https://doi.org/10.1016/j.enggeo.2022.106539.

[44] J.Z. Zhang, H.W. Huang, D.M. Zhang, K.K. Phoon, Experimental study of the coupling effect on segmental shield tunnel lining under surcharge loading and excavation unloading, Tunn. Undergr. Space Technol. 140 (2023) 105199, https://doi.org/10.1016/j.tust.2023.105199.

[45] J.Z. Zhang, Q.H. Jiang, D.M. Zhang, H.W. Huang, Z.Q. Liu, Influence of geological uncertainty and soil spatial variability on tunnel deformation and their importance evaluation, Tunn. Undergr. Space Technol. 152 (2024) 105930, https://doi.org/10.1016/j.tust.2024.105930.

[46] Z.Z. Wang, J.Z. Zhang, H.W. Huang, Interpreting random fields through the U-net architecture for failure mechanism and deformation predictions of geosystems, Geosci. Front. 15 (2024) 101720, https://doi.org/10.1016/j.gsf.2023.101720.

[47] D.M. Zhang, Z.Y. Ye, J.Z. Zhang, H.W. Huang, Displacement-based design of shield tunnel lining considering mobilized strength of undrained clay, Can. Geotech. J. 00 (2024) 1–17, https://doi.org/10.1139/cgj-2023-0362.

[48] D.M. Zhang, C. Nie, M. Chen, H. Huang, Y. Wu, Wireless tilt sensor based monitoring for tunnel longitudinal settlement: development and application, Measurement 217 (2023) 113050, https://doi.org/10.1016/j.measurement.2023.113050.

[49] Official documentation of Hyperledger Fabric. https://hyperledger-fabric.readthedocs.io/en/release-2.5/.

[50] J.Z. Zhang, K.K. Phoon, D.M. Zhang, H.W. Huang, C. Tang, Novel approach to estimate vertical scale of fluctuation based on CPT data using convolutional neural networks, Eng. Geol. 294 (2021) 106342, https://doi.org/10.1016/j.enggeo.2021.106342.