J Grid Computing (2024) 22:16
https://doi.org/10.1007/s10723-023-09739-x

RESEARCH

On-Chain and Off-Chain Data Management for Blockchain-Internet of Things: A Multi-Agent Deep Reinforcement Learning Approach

Y. P. Tsang · C. K. M. Lee · Kening Zhang · C. H. Wu · W. H. Ip

Received: 3 June 2023 / Accepted: 30 December 2023 / Published online: 20 January 2024

Abstract The emergence of blockchain technology has seen applications increasingly hybridise cloud storage and distributed ledger technology in the Internet of Things (IoT) and cyber-physical systems, complicating data management in decentralised applications (DApps). Because it is inefficient for blockchain technology to handle large amounts of data, effective on-chain and off-chain data management in peer-to-peer networks and cloud storage has drawn considerable attention. Space reservation is a cost-effective approach to managing cloud storage effectively, contrasting with the demand for additional space in real-time. Furthermore, off-chain data replication in the peer-to-peer network can eliminate single points of failure of DApps. However, recent research has rarely discussed optimising on-chain and off-chain data management in the blockchain-enabled IoT (BIoT) environment. In this study, the BIoT environment is modelled, with cloud storage and blockchain orchestrated over the peer-to-peer network. The asynchronous advantage actor-critic algorithm is applied to exploit intelligent agents with the optimal policy for data packing, space reservation, and data replication to achieve an intelligent data management strategy. The experimental analysis reveals that the proposed scheme demonstrates rapid convergence and superior performance in terms of average total reward compared with other typical schemes, resulting in enhanced scalability, security and reliability of blockchain-IoT networks, leading to an intelligent data management strategy.

Keywords Blockchain · Internet of Things · Data management · Deep reinforcement learning · Asynchronous advantage actor-critic (A3C) algorithm

Y. P. Tsang · C. K. M. Lee · K. Zhang
Department of Industrial and Systems Engineering, Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

C. H. Wu (✉) · W. H. Ip
Department of Supply Chain and Information Management, The Hang Seng University of Hong Kong, Shatin, Hong Kong
e-mail: jackwu@ieee.org

W. H. Ip
Department of Mech. Engg., The University of Saskatchewan, Saskatoon, Canada

1 Introduction

In recent years, the emergence of blockchain technology has led to significant changes in the design and development of application systems. For blockchain-driven applications, all data are decentralised to every system node. This is unfavourable for some data-rich applications, potentially substantially burdening the data storage and network bandwidth to decentralise a large-scale ledger in the peer-to-peer (P2P) network

[1]. This has prompted active exploration of the possibility of integrating blockchain and the Internet of Things (IoT) [2, 3] to enhance system scalability, reliability, and security. The rise of blockchain-IoT (BIoT) technology in recent years has driven the development of decentralised applications (DApps) built on the foundation of traditional applications and blockchain technology. Under the new cyber-physical system (CPS) paradigm, these DApps enable modern industrial solutions, such as supply chain [4] traceability and healthcare-decentralised identity. The increasing prominence of DApps designed and developed for the market demands is an effective data management strategy. Figure 1 graphically illustrates the generic topology of BIoT systems, which see P2P networks, IoT technologies, and cloud storage closely integrated.

At the peer nodes, smart devices represent client-side access to DApps that are interconnected with IoT devices for real-time data collection from physical objects and the environment. Based on the rules and logic behind DApps, application data are classified into on-chain and off-chain data that are managed in the P2P network and cloud database, respectively. This enables the minimal size of on-chain data – for example, data fingerprints – to be obtained to maintain the essential chain of data blocks (i.e., ledger) for data decentralisation. The burden on smart devices in terms of data storage can be reduced. In addition, on-chain data are associated with off-chain data stored in cloud databases via the edge and fog layers. Examples of off-chain data include original user data and files. The topology presented greatly improves system scalability and flexibility, such that the power of blockchain technology can be leveraged into end-user computing [5]. The practicality of DApps is further enhanced to cater for the demanding requirements of data-rich industrial applications. Despite the advantages of BIoT systems, there are some potential pitfalls in the design and development of BIoT systems. Because only on-chain data are decentralised in the P2P network, it is critical to consider single points of failure for off-chain data management. That is, if cloud storage shuts down, only the on-chain data – for example, hash values and Merkle tree roots – becomes meaningless without the off-chain data, making the overall design of DApps vulnerable to certain impacts of cyber-attacks [6]. Also, the effectiveness of data access is limited when relying on the centralised cloud due to network bandwidth and traffic restrictions. From the application design perspective, taking unlimited cloud storage space for granted is unreasonable due to the exponential increase in the data volume—consider, for example, the termination of unlimited free storage on Google Photos in June 2021 [7]. Elsewhere, cloud service platforms, such as Microsoft Azure, have recently encouraged resource reservation instead of creating resource demand in real-time, saving on system maintenance costs. Although existing studies have investigated resource

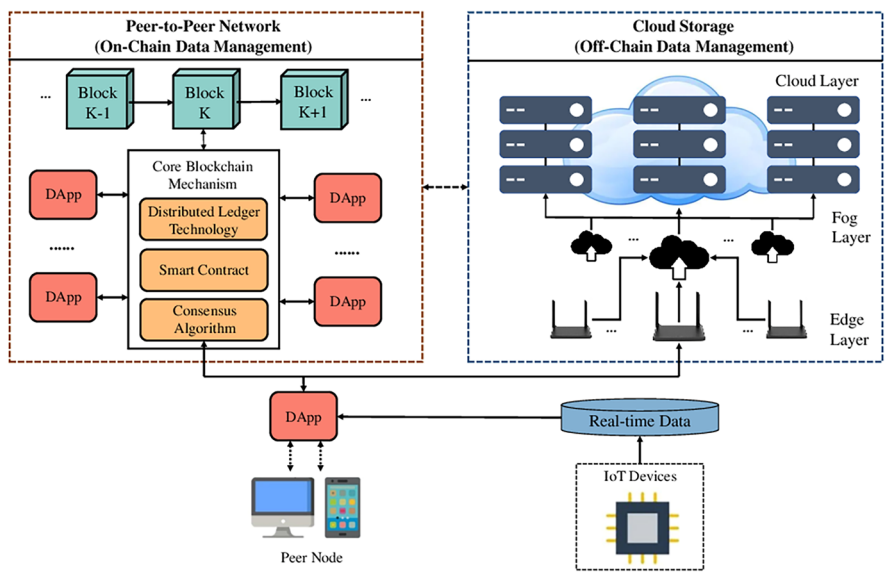

Fig. 1 Generic Topology of BIoT Systems

demand estimation based on typical forecasting methods, such as exponentially weighted moving averages [8] and autoregressive integrated moving averages [9], they cannot efficiently address volatile data management demand in a complex network environment. Therefore, studies are increasingly exploring machine learning applications for resource management in the cloud environment [10, 11].

In the data management of DApps, the issues mentioned above, including single point failure, insufficient storage space and network bandwidth limitation, must be addressed simultaneously. This research proposes an integrated BIoT system environment for DApps that can orchestrate cloud storage and blockchain over the P2P network to revamp current BIoT systems methods further. By leveraging the power of artificial intelligence (AI), it aims to enhance the scalability of the blockchain-IoT network through intelligent data management. The contributions can be summarised as follows.

- We propose an on-chain and off-chain data management framework for BIoT. In the proposed system, to achieve data replication in the P2P network in a manner distinct from complete data decentralisation, storage spaces shared by peer nodes are utilised to manage the fragmented off-chain data, ideally inhibiting unnecessary data rearrangement. This reduces the risk of data leakage and loss due to a single point of failure.
- The proposed system model sees a deep-reinforcement learning agent exploited by applying the asynchronous advantage actor-critic (A3C) algorithm to support decisions on data packing, cloud storage reservation, and off-chain data replication in the P2P network, with offloading decisions, power allocation, block size, and block intervals jointly optimised.
- We evaluate the performance and effectiveness of the proposed strategy, in which three experimental schemes are considered: (i) identical data size, (ii) identical degree of data importance, and (iii) fixed block size and block interval. The simulation results show that the proposed framework highlights the value of intelligent on-chain and off-chain data management over blockchain-IoT networks, which encourages scalable and sustainable DApp development within the context of the CPS paradigm.

This paper is organised as follows. Section 2 reviews (i) contemporary trends in BIoT design and development and (ii) state-of-the-art data management methods. Section 3 formulates the BIoT system's on-chain and off-chain data management modules, and the A3C algorithm is customised to support the sequential decision-making process in Section 4. The simulation experiments and results appear in Section 5, and Section 5.3 discusses the value and practical implications of this research before Section 6 draws conclusions and indicates future possibilities.

2 Literature Review

This section reviews the background of BIoT systems to summarise the major components and data management practices. Elsewhere, existing data management practices are synthesised as a foundation for addressing the research gap of interest.

2.1 Contemporary BIoT Trends and Development

Conventional IoT technologies do not guarantee data traceability and accountability, and the exposure to cybersecurity risks associated with using a centralised server cannot be effectively eliminated. This has prompted the emergence of the BIoT paradigm, which overcomes the pitfalls associated with relying entirely on IoT technologies for the design and development of smart and intelligent solutions [12–16]. Furthermore, the fusion of blockchain with IoT produces more opportunities to reveal the power of blockchain technology in various industries for the development of specific data-rich applications designed for the establishment of CPS.

However, although existing studies demonstrate the trends and intellectual foundation of BIoT, BIoT systems are yet to mature. The effective integration of BIoT with the CPS has received considerable attention from academics and industrial practitioners interested in DApp development. More specifically, despite the critical need to balance decentralised and centralised control mechanisms, this remains under-explored in terms of the allocation of network resources and the scheduling of network traffic. For instance, blockchain-based product traceability and authentication require the data storage

of comprehensive supply chain information, which includes, for example, production, distribution, and sales [17]. Although the number of nodes in blockchain-based traceability solutions is not comparable to public blockchain technologies (e.g., bitcoin), the required throughput capacity must be relatively high to capture traceability data effectively as transaction records. Furthermore, various consensus algorithms, including proof of authority and Raft, have been designed and applied with corresponding fault tolerance levels. The fault tolerance of BIoT systems depends on the type of consensus algorithm. For the IBFT and QBFT under the proof of authority algorithm, a maximum of one-third of malicious nodes can be tolerated across the entire system network. For the Raft algorithm, used in Amazon Managed Blockchain and IBM Blockchain Platform, a maximum of $(n-1)/2$ crashed nodes can be tolerated by a total of n nodes.

Regarding the software development process, Marchesi et al. [18] have proposed agile blockchain DApp engineering to outline data flows of smart contracts and real-time user interactions under the Ethereum protocol. The front-end systems are interconnected with the blockchain and back-end server, with the nodes and their storage in the P2P network managed by blockchain-based virtual machines (VMs), such as the Ethereum VM. Unlike conventional VM schedulers in cloud storage, blockchain-based VMs play an essential role in allocating storage volume and assisting block decentralisation over the P2P network. Consequently, the system architecture of DApps is far more complicated than that of traditional applications because the need to consider the P2P network and cloud storage together demands that the complex data management of BIoT systems balances decentralisation and scalability.

2.2 State-of-the-Art Data Management Methods

For many, resource allocation in computer system networks represents the foundation of effective data management and includes, for example, optimal data storage and load balancing. Storage service providers deploy the VM schedulers – the core of VM monitors – in cloud storage to support load prediction, data scheduling, and migration, enabling effective management of physical machines [19]. Each physical machine can implement multiple VMMs, which can collect three major resource allocation statuses: (i) the resource demand history of VMs, (ii) the capacity and load history of PMs, and (iii) the current assignment of VMs to PMs. These inputs can be analysed to support load prediction and VM layout optimisation and, in return, maximise resource utilisation and satisfy all resource demands [20, 21]. Most extant studies have examined the bin packing algorithm to allocate and provide optimal resources for data storage [19]. It is commonly known that the bin packing problem for resource allocation is NP-hard. The asymptotic performance ratio for an algorithm A to perform bin packing for a given list L of the input sequence of n items is presented as Eq. (1), where $A(L)$ and $OPT(L)$ denote the number of bins determined by algorithm A and the optimal algorithm, respectively. Furthermore, the online bin packing algorithm has been developed to address the challenge of handling variable item sizes in an online regime – consider, for example, Next Fit for $AP(A) = 2$, First Fit for $AP(A) = 1.7$, and Best Fit for $AP(A) = 1.7$ [22]. However, the complexity of the computer system network renders this classical and simple algorithm unsuitable for data management in a BIoT system.

Data fragmentation has been introduced to enhance system security and improve storage efficiency to facilitate storage space utilisation from the system perspective [23, 24]. Data packages can be dynamically fragmented and fit into storage bins by defining fragmentation properties, such as chunk size. Because every storage bin (i.e., the hard drive) has limitations on read/write performance, putting all VM data into a large storage bin can affect the user experience of data query and creation. In real-life situations, a sufficient number of storage bins should be guaranteed via cloud storage reservation to prevent the suspension of service and improve cost-effectiveness. Although managing VM data into a number of data chunks can enhance query performance and system security, the possibility of a single point of failure in the cloud environment cannot be fully avoided. Accordingly, decentralised cloud storage based on blockchain technology has been actively explored in the context of cybersecurity in cloud storage [25]. The data can be replicated in the computer network according to their importance for the entire application system. Particularly for BIoT systems, such data replication can be considered a hybrid approach that

strikes a balance between full data centralisation and decentralisation. Consequently, redundant storage spaces in the P2P network can be fully utilised to save extra copies of the off-chain data, without which only on-chain data (e.g., hash keys) remain available if a BIoT system is shut down in response to a cyber attack. An effective policy that allocates sufficient resources in BIoT systems represents an unprecedented development in balancing cybersecurity and resource allocation.

This literature review reveals that the BIoT paradigm represents a promising contemporary shift in the direction of enhanced system security and business synergy via data decentralisation. Accordingly, the near future promises the design and development of increasing numbers of BIoT systems. Differing from past cloud solutions, both on-chain and off-chain data should be effectively managed in terms of resource allocation, load balancing, and system security. Table 1 summarises the comparison of relevant studies to highlight the research motivation of this study. On the one hand, an effective cloud storage reservation should be implemented to maintain sufficient and timely storage volume and optimise data packing into storage bins. On the other hand, the concept of the decentralised cloud should be extended into the P2P network to eliminate single points of failure and enhance system security and reliability. Redundant space in P2P networks and cloud storage should be fully utilised, and frequent data re-packing should be inhibited. An effective data management policy should be established for the recent BIoT topology to achieve a secure and resource-efficient application system to sustain the exploitation of the BIoT system. This means that this study is motivated to construct an intelligent agent for BIoT systems, with the potential for DRL explored to adapt to the dynamic computing environment.

3 System Framework and Model

This section proposes and formulates the system model of the BIoT, which it describes in terms of two major aspects: (i) the system framework and (ii) key modules of the system model. Regarding the whole architecture, IoT devices (IoTDs) are distributed across different locations. They can collect diverse real-time data, which can be stored on the blockchain or transmitted directly to the cloud. Small base stations (SBSs) containing several connected IoTDs generate blocks in the P2P network. This hybrid topology is defined in Fig. 2; the black, coloured and hollow circles represent the cloud server, SBSs and IoTDs, respectively. This structure significantly enhances system scalability and flexibility since each part offers unique functionalities and advantages and enables the system to adapt to varying scales and application scenarios for data management.

3.1 Overall System Framework

Figure 3 illustrates the proposed framework deployed in the BIoT system environment. Peer nodes $N = \{1, 2, \ldots, n\}$ participating in the IoTD system can create time-series information (e.g., shipment updates of the traceability system and health status of the e-health record) at time $T = \{1, 2, \ldots, t\}$ [16]. For some nodes, $\exists n \in N$, available storage volumes are heterogeneously provided and shared in the P2P network to achieve data replication. In the BIoT system

Table 1 Comparison with existing research studies

References	[22]	[18]	[19]	[14]	This study
Off-chain storage	✓		✓		✓
On-chain storage				✓	✓
Online requests	✓		✓		✓
Cost minimisation			✓		✓
System architecture		✓			✓
Method	k-Binary algorithm	-	VISBP algorithm	IBM-DKG	A3C algorithm

*Remarks: VISBP denotes Bin packing with variable item size; IBM-DKG denotes Intelligent blockchain management for distributed knowledge graph matching; A3C algorithm denotes the asynchronous advantage actor critic algorithm

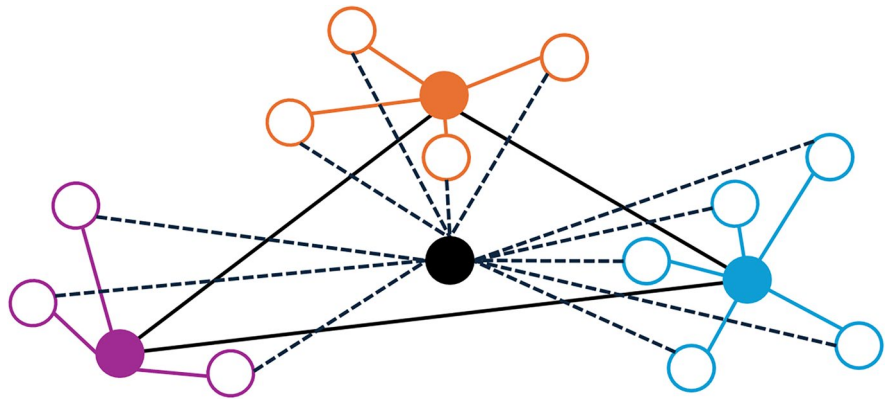

Fig. 2 The hybrid topology of the proposed system

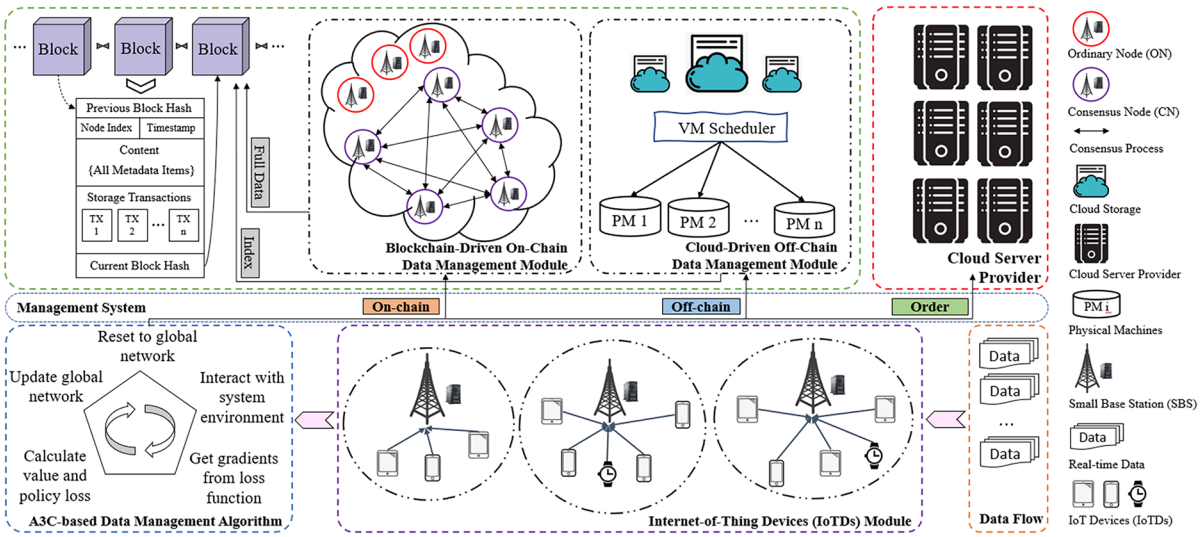

Fig. 3 The Framework of the Proposed On-Chain and Off-Chain Data Management System

environment, uploaded transactions are verified by specific consensus algorithms, meaning that decisions concerning transaction data management involve two choices: (i) on-chain data and (ii) off-chain data. For on-chain data management at time t, the blockchain is appended with the new on-chain data and decentralised in the P2P network, with the size of the on-chain data defined as $D^{on} = \sum_{t \in T} d_t^{on}$ and $d^{on} \in (0, 1]$. Meanwhile, the off-chain data are transmitted to the cloud for long-term storage, where the VM scheduler manages the system load, scheduling, and data migration for a certain number of homogeneous physical machines (PMs) with a normalised capacity of 1. The size of off-chain data is subsequently modelled as $D^{off} = \sum_{t \in T} d_t^{off}$ and $d^{off} \in (0, 1]$, with the VM scheduler required to pack multiple sets of off-chain data at time t. The proposed system environment restricts the number of pending PMs for executing data storage, meaning that additional PMs should be reserved based on the current data storage utilisation level. Three major classes describe the nature of PMs: (i) active, (ii) reserved, and (iii) inactive for PMs that are in use, reserved, and unusable, respectively. The appropriate number of PMs should be reserved to operate DApps rather than assuming that all PMs in the data centres can be utilised and deployed in real-time. Upon reserving PMs, additional storage bins are made available to BIoT systems. From the perspective of the entire computer network, two sets of bins are defined: (i) homogeneous bins in the cloud storage

and (ii) heterogeneous bins in the P2P network. The re-arrangement of the packed data should be inhibited when packing the off-chain data into the two storage bins to avoid a sudden upsurge in network traffic and demand for a larger bandwidth to achieve data-packing optimisation. Data fragmentation is preferred to enhance storage bin utilisation, but data query efficiency may be affected because re-packing fragmented data takes considerable time. The query speed limitation of hard drives (e.g., 50–250 MB/s in 4 K read/writes of solid-state drives) renders optimal data fragmentation decision-making essential to smoothen the data access of system operations.

This complicates the decision-making process for packing new data, re-locating packed data, and reserving additional PMs in the context of the BIoT system environment. Alleviating these challenges involves exploring the DRL approach to support data management decisions. Upon achieving the convergence of the DRL process, the intelligent agent can be embedded in the system environment such that the resources of the cloud and P2P storage are fully utilised to balance the cybersecurity, scalability, and cost-effectiveness of the BIoT system.

3.2 Key Modules of the BIoT System Model

Building on the overview of the BIoT system environment, the key modules of the system model are illustrated using mathematical formulations, in which the system variables are summarised in Table 2. This paper assumes that peer nodes in the network transmit data along orthogonal spectrums, keeping the system interference-free [26].

3.2.1 IoTDs Model

On one hand, regarding the on-chain and off-chain data transmission in the proposed network, the sets of IoTDs and SBSs are denoted as $\mathcal{X} = \{1, 2, ..., x, ..., X\}$

Table 2 Notation of system variables

Symbol	Description
$\mathcal{X}$	The set of all connected IoT devices
$\mathcal{Y}$	The set of all small base stations
$\mathcal{Z}$	The set of all homogeneous physical machines in the cloud server
$\mathcal{D}$	The set of data flow
$D_q(1), L_q$	The size of the corresponding tasks
$D_q(2), M_q$	The importance degree of tasks
B	The total bandwidth
$g_{x,y}^{onc}$	The channel gain from the IoTD x to the SBS y
$p_{x,y}^{onc}$	The transmit power from the IoTD x to the SBS y
p_{max}^{onc}	The allowed maximum transmit power allocation of each IoTD
N_y	The total number of IoTDs connected with SBS y
σ_{onc}^2	The on-chain noise power in the transmission
σ_{off}^2	The off-chain noise power in the transmission
$\gamma_{x,y}^{onc}$	The channel-to-interference plus-noise ratio from IoTD x to SBS y
γ_x^{offc}	The channel-to-interference plus-noise ratio from IoTD x to the cloud server
$\upsilon_{x,y}^{onc}$	The transmission data rate from IoTD x to SBS y
υ_x^{offc}	The transmission data rate from IoTD x to SBS y
$\mathcal{K}_x^{sum}$	The total transmitted energy consumption from IoTD x to the SBS or cloud server
$\mathcal{T}_x^{sum}$	The total transmit time from IoTD x to the SBS or cloud server
$\rho_R^B(t)$	The set of available resources in a short-notice time period t
$\varpi_S^B(t)$	The set of stales of each node in a short-notice time period t
$I^B(t)$	The interval (in seconds) in a short-notice time period t
$L^B(t)$	The size (in MB) of blocks in a short-notice time period t
$\wp(t)$	The transaction throughput in a short-notice time period t

and $\mathcal{Y} = \{1, 2, ..., y, ..., Y\}$, respectively. There are $|\mathcal{X}|$ IoTDs installed Dapps and $|\mathcal{Z}|$ SBSs, which are integrated with edge servers. Note that for the on-chain part, Y_B block producers are chosen from set $\mathcal{Y}$ following the corresponding consensus mechanism, which is stated specifically in Sect. 3.2.2. Due to the low storage and poor computing capability of IoTDs, tasks will be transmitted to SBSs or CS via a wireless network. In the CS as Z_0, there are homogeneous PMs, denoted as $\mathcal{Z} = \{1, 2, ..., z, ..., Z\}$. All the PMs are deployed in a close location, with the time for transmitting data to different PMs treated as the same. Similar to [27], the transmission model utilises the universal frequency reuse scheme, meaning that the on-chain network adopts the same radio and the off-chain network adopts the other same radio. For the arrival of data flow, it is assumed that the data will be imported into the system sequentially, denoted as the set $\mathcal{D} = \{D_1, D_2, .., D_q, ..., D_Q\}$, where D_q represents the combination of the size and the degree of importance of each task, with each element $D_q = [L_q, M_q]^T$, where $D_q(1) = L_q(L_q > 0)$ refers to the size of the corresponding task and $D_q(2) = M_q(M_q \in [0, 1])$ refers to the importance degree of tasks. B^{onc} and B^{offc} represent the on-chain and off-chain bandwidth, and the total bandwidth is B Hz, where $B = B^{onc} + B^{offc}$.

Regarding on-chain data transmission, let the channel gain $\mathcal{G}^{onc} = (g^{onc}_{x,y})$ and transmit power $\mathcal{P}^{onc} = (p^{onc}_{x,y})$ in the system, where $p^{onc}_{x,y}$ and $g^{onc}_{x,y}$ refer to the matching parameters from IoTD x to SBS y, respectively. Building on [27], the channel-to-interference plus-noise ratio from IoTD x to SBS y can be represented by Eq. (1), where p^{onc}_{max} is the allowed maximum transmit power allocation of each IoTD and σ^2_{onc} denotes the noise power in the transmission.

$$\gamma^{onc}_{x,y} = \frac{g^{onc}_{x,y} p^{onc}_{x,y}}{\sigma^2_{onc} + \sum_{i \in \mathcal{X}-\{x\}, j \in \mathcal{Y}-\{y\}} g^{onc}_{i,j} p^{onc}_{max}} \tag{1}$$

Subsequently, it is assumed that the total number of IoTDs connected to SBS y is denoted as N_y. Then, the bandwidth of each IoTD will be B^{onc}/N_y. This enables the transmission data rate from IoTD x to SBS y to be represented by Eq. (2):

$$\upsilon^{onc}_{x,y} = \frac{B^{onc}}{N_y} log_2(\gamma^{onc}_{x,y} + 1) \tag{2}$$

Thus, the transmit power from IoTD x to SBS y is given in Eq. (3) by joining Eqs. (1) and (2):

$$p^{onc}_{x,y} = \frac{\sigma^2_{onc} + \sum_{i \in \mathcal{X}-\{x\}, j \in \mathcal{Y}-\{y\}} g^{onc}_{i,j} p^{onc}_{max}}{g^{onc}_{x,y}} (2^{\frac{\upsilon^{onc}_{x,y} N_y}{B^{onc}}} - 1) \tag{3}$$

Let $\mathcal{V}^{onc} = (\upsilon^{onc}_{x,y})$ be the vector of the transmit rate of the on-chain system from IoTD x to SBS y and the data size from IoTD x be denoted as D^{onc}_x. Subsequently, the time for transmitting data can be represented as $t^{Nonc}_{x,y} = D^{onc}_x / \upsilon^{onc}_{x,y}$, enabling the transmission of energy consumption $\mathcal{K}^{Nonc} = (k^{Nonc}_{x,y})$ from IoTD x to SBS y to be given by Eq. (4):

$$\begin{aligned} k^{Nonc}_{x,y} &= p^{onc}_{x,y} t^{Nonc}_{x,y} \\ &= \frac{(\sigma^2_{onc} + \sum_{i \in \mathcal{X}-\{x\}, j \in \mathcal{Y}-\{y\}} g^{onc}_{i,j} p^{onc}_{max}) D^{onc}_x}{\upsilon^{onc}_{x,y} g^{onc}_{x,y}} (2^{\frac{\upsilon^{onc}_{x,y} N_y}{B^{onc}}} - 1) \end{aligned} \tag{4}$$

Regarding off-chain data transmission, similar to the on-chain network, let $\mathcal{G}^{offc} = (g^{offc}_x)$ and $\mathcal{P}^{offc} = (p^{offc}_x)$. Although there are many PMs in the CS Z_0, they are deployed in the same position. Therefore, the transmit power, the channel gain, and the network delivery time should be considered approximately equal, with g^{offc}_x and p^{offc}_x only related to IoTDs. This is given by Eq. (5), where p^{offc}_{max} is the maximum transmit power allocation and σ^2_{offc} is the noise power in the transmission of each IoTD:

$$\gamma^{offc}_x = \frac{g^{offc}_x p^{offc}_x}{\sigma^2_{offc} + \sum_{i \in \mathcal{X}-\{x\}} g^{offc}_i p^{offc}_{max}} \tag{5}$$

Subsequently, the average frequency bandwidth is denoted as $B^{offc}/|\mathcal{X}|$ for each IoTD, because all the devices can be associated with the CS Z_0. Therefore, the transmission data rate from IoTD x to the CS Z_0 can be given by Eq. (6):

$$\upsilon^{offc}_x = \frac{B^{offc}}{|\mathcal{X}|} log_2(\gamma^{offc}_x + 1) \tag{6}$$

Accordingly, the transmit power from IoTD x to CS Z_0 is given in Eq. (7) by joining Eqs. (5) and (6):

$$p^{offc}_x = \frac{\sigma^2_{offc} + \sum_{i \in \mathcal{X}-\{x\}} g^{offc}_i p^{offc}_{max}}{g^{offc}_x} (2^{\frac{\upsilon^{offc}_x |\mathcal{X}|}{B^{offc}}} - 1) \tag{7}$$

Let $\mathcal{V}^{offc} = (\upsilon^{offc}_x)$ be the vector of the transmit rate of the off-chain system from IoTD x to CS Z_0, with the data size from IoTD x denoted as D^{offc}_x. Next,

the time for transmitting data can be represented as $t_x^{Noffc} = D_x^{offc}/\upsilon_x^{offc}$, enabling the transmission of energy consumption $\mathcal{K}^{Noffc} = (k_x^{Noffc})$ from IoTD x to CS Z_0 to be given by Eq. (8):

$$k_x^{Noffc} = p_x^{offc} t_x^{Noffc} = \frac{(\sigma_{offc}^2 + \sum_{i \in \mathcal{X} - \{x\}} g_i^{offc} p_{max}^{offc}) D_x^{offc}}{\upsilon_x^{offc} g_x^{offc}} (2^{\frac{\upsilon_x^{offc}|\mathcal{X}|}{B^{offc}}} - 1) \tag{8}$$

On the other hand, for on-chain and off-chain computation, the process time can be represented by Eq. (9), with the data sent to SBS y from IoTD x needing to be stored in the read-only memory, meaning that W_x and $f_{x,y}$ (in cycles/s) represent CPU cycles of process unit data and CPU-cycle frequency prescribed by the equipment requirements from IoTD x to y th SBS, respectively.

$$t_{x,y}^{Conc} = \frac{W_x D_x^{onc}}{f_{x,y}} \tag{9}$$

Building on [28], the effective switched capacitance of the CPU in the y th SBS can be denoted as η_y^{SBS}, and the power of processing the data from IoTD x to the y th SBS will be $p_{x,y}^{SBS} = f_{x,y}^3 \eta_y^{SBS}$[29]. Accordingly, the energy consumed in the y th SBS for processing data from IoTD x can be represented by Eq. (10), where $\mathcal{K}^{Conc} = (k_x^{Conc})$.

$$k_{x,y}^{Conc} = p_{x,y}^{SBS} t_{x,y}^{Conc} = f_{x,y}^2 \eta_y^{SBS} W_x D_x^{onc} \tag{10}$$

However, for off-chain computation, when computing tasks are transmitted to CS Z_0, the CS needs to allocate the proper PM to store the task of IoTD x, which results in different CPU-cycle frequencies for processing, denoted as $f_{x,z}$ from IoTD x to z th PM. This enables the time that PM z processes the data of IoTD x to be given by Eq. (11):

$$t_{x,z}^{Coffc} = \frac{W_x D_x^{offc}}{f_{x,z}} \tag{11}$$

Furthermore, the effective switched capacitance of the CPU in the z th PM can be denoted as η_z^{PM} [28], and the power of processing the data from IoTD x to the z th PM [29] will be $p_{x,z}^{PM} = \eta_z^{PM} f_{x,z}^3$. Consequently, the energy consumed by the z th PM to process the data of IoTD x can be given by Eq. (12), where $\mathcal{K}^{Coffc} = (k_x^{Coffc})$:

$$k_{x,z}^{Coffc} = p_{x,z}^{PM} t_{x,z}^{Coffc} = f_{x,z}^2 \eta_z^{PM} W_x D_x^{offc} \tag{12}$$

In brief, on-chain energy from IoTD x to y th SBS and off-chain energy from IoTD x to z th PM can be summarised by Eqs. (13) and (14):

$$k_{x,y}^{onc} = k_{x,y}^{Nonc} + k_{x,y}^{Conc} \tag{13}$$

$$k_{x,z}^{offc} = k_x^{Noffc} + k_{x,z}^{Coffc} \tag{14}$$

Meanwhile, the on-chain time from IoTD x to y th SBS and the off-chain time from IoTD x to z th PM can be summarised by Eqs. (15) and (16):

$$t_{x,y}^{onc} = t_{x,y}^{Nonc} + t_{x,y}^{Conc} \tag{15}$$

$$t_{x,z}^{offc} = t_x^{Noffc} + t_{x,z}^{Coffc} \tag{16}$$

It is assumed that there are $\mathcal{E}_x$ affairs in each IoTD x, $\mathfrak{E}_{IoTD} = \{\mathcal{E}_x | \forall x \in \mathcal{X}\}$. The energy consumption and execution time of each affair in IoTD x can be denoted as $k_{x,y}^{onc(j)}$, $k_{x,z}^{offc(j)}$, $t_{x,y}^{onc(j)}$ and $t_{x,z}^{offc(j)}$, $\forall j \in \{1, 2, ..., \mathcal{E}_x\}$. Next, it is denoted that $\mathfrak{T}_{IoTD}^{Aff} = \{I_x | \forall x \in \mathcal{X}\}$ (as the user association variable set) and $I_x = \left\{ i_j^x | \forall j \in \mathcal{E}_x \right\}$, where $i_j^x \in \{0, 1\}$, $\forall x \in \{1, 2, ..., X\}$, $\forall j \in \{1, 2, ..., \mathcal{E}_x\}$. When $i_j^x = 1$ (on-chain), the IoTD x transmits the data to the SBS and when $i_j^x = 0$ (off-chain), the IoTD x transmits the data to the z th PM in CS Z_0. Consequently, the overall energy consumption and the total processing time in the proposed system for the data flow from each IoTD x can be represented by Eqs. (17) and (18), respectively.

$$\mathcal{K}_x^{sum} = \sum\nolimits_{j \in \{1,2,...,\mathcal{E}_x\}, \mathcal{E}_x \in \mathfrak{E}_{IoTD}} \left\{ i_j^x k_{x,y}^{onc(j)} + \left(1 - i_j^x\right) k_{x,z}^{offc(j)} \right\} \tag{17}$$

$$\mathcal{T}_x^{sum} = \sum\nolimits_{j \in \{1,2,...,\mathcal{E}_x\}, \mathcal{E}_x \in \mathfrak{E}_{IoTD}} \left\{ i_j^x t_{x,y}^{onc(j)} + \left(1 - i_j^x\right) t_{x,z}^{offc(j)} \right\} \tag{18}$$

3.2.2 *Blockchain-Driven On-Chain Data Management Module*

Every node from the proposed system can collect transactions for the blockchain-driven module. Due to the impossible triangle in the blockchain network (including public chain, alliance chain, private

chain, and so on), various methods of constructing consensus can be implemented to equalise security and performance, including the value of stales, confidence level and remaining resources. A consensus algorithm for blockchain is used in a blockchain network to reach consensus and confirm the validity of transactions. A consensus algorithm aims to achieve a decentralised trust mechanism by getting participants to agree. Common blockchain consensus algorithms include PoW (Proof of Work), PoS (Proof of Stake), PBFT (Practical Byzantine Fault Tolerance) and so on. In order to reduce the computational pressure of edge servers and to better match real-world scenarios, we adapt the delegated Byzantine Fault Tolerance 2.0 (dBFT 2.0) algorithm [30] adopted to achieve peer node consistency, as Fig. 4 shows. Unlike dBFT, dBFT 2.0 proposes a three-stage consensus mechanism and a recovery mechanism, enhancing the security and robustness of blockchain networks [31].

Suppose that the available resource and the value of stakes of the peer node y at a short-notice time period t are denoted as $\rho_y(t)$ and $\varpi_y(t)$ in the network, where $\rho_R^B(t) = \{\rho_1(t), \rho_2(t), \ldots, \rho_Y(t)\}$ and $\varpi_S^B(t) = \{\varpi_1(t), \varpi_2(t), \ldots, \varpi_Y(t)\}$, respectively. Assuming that there is a first-in-first-out buffer for SBSs to store the data not processed in the queue, the recurrence formula for processing the data queue deriving from IoTDs on the SBS y at the next short-notice time shot $t+1$ can be denoted as $r_y(t+1) = max\left\{0, r_y(t) + t_{x,y}^{Conc} D_x^{onc}/f_{x,y} - f_{x,y}\eta_y^{SBS}\right\}$. Subsequently, the resources available in the SBS y at a short-notice period can be represented as $\rho_y(t) = max\{r_{sum}(t) - r_y(t), \rho_{min}(t)\}$, where $r_{sum}(t) = \sum_{y\in\mathcal{Y}} r_y(t)$ and $\rho_{min}(t) = min\{\rho_1(t), \rho_2(t), \ldots, \rho_Y(t)\}$.

In this system model, it is assumed that there are P block producers in peer nodes playing the role of generating blocks [32]. The interval (in seconds) and size (in MB) of blocks are denoted as $I^B(t)$ and $L^B(t)$ at a short-notice period t, where $I^B(t) \in [I_m, I_M]$ and $L^B(t) \in [L_m, L_M]$, respectively. In this consensus algorithm, a maximum of q fault nodes can be tolerated, supposing that P nodes participate in consensus decision-making, where $P \geq 3q + 1$. The leader of nodes in the blockchain network is regarded as the speaker $\mathfrak{f}$, and the others are members. The new block will be broadcast from the speaker to members, who vote on this decision after receiving the proposal. All nodes can pass the proposal verification provided that the number of votes in favour is greater than or equal to $P - q$. For block verification, the parameter p^s is assumed to be the probability of successfully passing the verification. Meanwhile, the speaker is identified

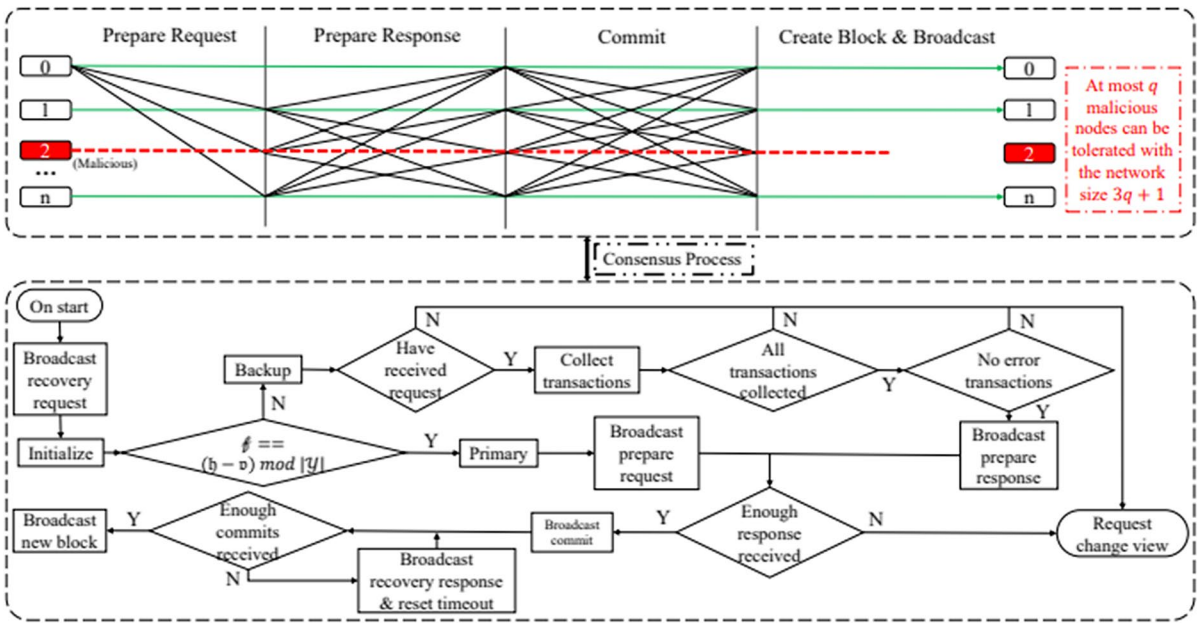

Fig. 4 Process of the dBFT 2.0 Consensus Algorithm

by Formula (19), where $\mathfrak{h}$ and $\mathfrak{v}$ are the height and view number of blocks.

$$\mathfrak{f} = (\mathfrak{h} - \mathfrak{v})\ mod\ |\mathcal{Y}| \tag{19}$$

A round of consensus in the dBFT 2.0 can be divided into four phases: (i) prepare request, (ii) prepare response, (iii) commit, and (iv) create block and broadcast. At the beginning of the first phase, the speaker broadcasts the prepare-request message to all the members to start a new round of consensus. Then, in the prepare-response phase, delegates broadcast the message about the successful data collection for block creation. In the third phase, the above messages have been broadcast to a sufficient number of validators for the block verification process. Finally, the new block is produced and broadcast by validators after verification, and then the blockchain system continues to start the next round of the consensus process.

During the consensus procedure, the time cost in one round can be denoted as $\rho_C^B(t)$. For model-building convenience, the consensus is formulated as two sections, the propagation section and the verification section [33], and the corresponding time costs are represented by $\rho_P^B(t)$ and $\rho_V^B(t)$ (in a short-notice period t). Naturally, the total time cost $\rho_C^B(t) = \rho_P^B(t) + \rho_V^B(t)$. Building on [32], latency time to finality in the blockchain module can be expressed by $\rho_T^B(t) = \rho_C^B(t) + I^B(t)$. Subsequently, the transaction throughput [32] of the blockchain system can be given by Eq. (20), where ξ is the average size:

$$\wp(t) = \lfloor L^B(t)/\xi \rfloor I^{B(-1)}(t) \tag{20}$$

3.2.3 Cloud-Driven Off-Chain Data Management Module

Cloud clustering involves leasing several working nodes (VMs) from the cloud to store corresponding tasks. It is supposed that there are D data affairs in the cloud server in the time interval. For data d_t^{off}, $d_t^{off} \in \left\{ d_1^{off}, d_2^{off}, \ldots, d_D^{off} \right\}$, the average request rate of the system for data affair d_t^{off} in caches, at a time slot t can be represented as in Eq. (21), where index t denotes the t th affair in the server buffer. It is assumed that the data flow enters the buffer following the Poisson process with the parameter β [34]. Similar to [35], the probability of a data affair chosen to store obeys the Zipf-like distribution, which can be denoted as $1/\mu(d_t^{off})^{\alpha}$, where $\mu = \sum_{t=1}^{D} 1/(d_t^{off})^{\alpha}$ and $\alpha (0 < \alpha < 1)$ is the Zipf slop.

$$\gamma_d(t) = \frac{\beta}{\mu(d_t^{off})^{\alpha}} \tag{21}$$

In the cloud-driven module, the corresponding data are cyclically stored by caches in the cloud server. It is assumed that a random parameter τ_i denotes whether the server requests data affair d_t^{off} in the cache. Building on [34], the state S_D of the cache can be considered the Markov chain, in which $\tau_i \in S_D$, $S_D = 1$, and $S_D = 0$ indicate corresponding data affairs situated in the cache (or not). Subsequently, the state transition probability matrix (STPM) of data affairs d_t^{off} at a time slot t in the cloud can be denoted as in Eq. (22), where $\eta_{\rho_d \vartheta_d}(t) = P(\tau_d(t+1) = \vartheta_d | \tau_d(t) = \rho_d)$, and ϑ_d, $\rho_d \in S_D$.

$$\Theta_d(t) = \left[\eta_{\rho_d \vartheta_d}(t) \right]_{2\times 2} \tag{22}$$

There are two types of data storage in the server cache: finite type and infinite type. In the first category, all the data affairs derived from IoTDs can be stored integrally and not removed until they exceed the allotted time. During this process, the retention time – when data affairs exist in the server cache – can be modelled as an exponential distribution with the mean $1/\varepsilon_d$. The corresponding STPM of the server cache can be expressed by Eq. (23) [34]:

$$\Omega_d^0 = \begin{bmatrix} -\gamma_d & \gamma_d \\ \epsilon_d & -\epsilon_d \end{bmatrix} \tag{23}$$

In the second category, STPM can be formulated by various cloud server cache replacement strategies. As in most cases, the least recently used policy is utilised in this proposed module, with the matrix represented in Eq. (24) [34], where $\tau_d = \beta - \gamma_d$. Note that CPU cores are assumed to be the available resources of the cloud server.

$$\Omega_d^1 = \begin{bmatrix} -\gamma_d & 0 & \cdots & 0 & 0 & \gamma_d \\ \tau_d + \epsilon_d & -\beta - \epsilon_d & \cdots & 0 & 0 & \gamma_d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \epsilon_d & 0 & \vdots & \tau_d & -\beta - \epsilon_d & \gamma_d \\ \epsilon_d & 0 & \vdots & 0 & \tau_d & -\tau_d - \epsilon_d \end{bmatrix} \tag{24}$$

4 A3C Algorithm for On-Chain and Off-Chain Data Management

Due to the non-deterministic polynomial-time (NP)-hardness of the cloud storage optimisation problem, the data management in BIoT systems (i.e., managing cloud and P2P storage simultaneously) is deemed to be NP-hard, such that the optimisation problem cannot be solved in polynomial time. The problem is formulated as a Markov decision process in the proposed data management system, where variable data sizes, data importance, block size, and block interval are considered for deriving optimal data management decisions. Traditional heuristic algorithms may struggle to effectively handle such high-dimensional and dynamic decision spaces. Additionally, BIoT environments are inherently uncertain, with data arrival patterns and system dynamics that can change over time. DRL is well-suited for solving problems where decisions must be made sequentially, as it can capture the sequential dependencies, learn from interactions with the environment and adapt strategies to handle uncertainty effectively. Therefore, the DRL approach is applied to build numerous intelligent agents, achieving decentralised autonomous operations (DAO) in data management.

4.1 State Variable and Space

Based on the definition of the Markov chain, a series of epochs t, $t \in \mathbb{N}^+$ can be the decision-making moment, such that the system state space can be defined following t. The state can be divided into five parts of the proposed data management system. The first part is the data storage situation, denoted $D(t) = \{D_1(t), D_2(t), ..., D_{(Y+Z)}(t)\}$, $D_i(t) \in \mathbb{N}$, and $|D(t)| = |\mathcal{Y}| + |\mathcal{Z}|$. $D_i(t)$ represents the number of data affairs in the i th SBS or PM of the system. The other parts are the transmit power $\mathfrak{P}(t) = \{\mathcal{P}^{onc}(t), \mathcal{P}^{offc}(t)\}$, wireless channel conditions $\mathfrak{G}(t) = \{\mathcal{G}^{onc}(t), \mathcal{G}^{offc}(t)\}$, the available resources of the SBS $\rho_R^B(t) = \{\rho_1(t), \rho_2(t), \ldots, \rho_Y(t)\}$, and the number of stakes $\varpi_S^B(t) = \{\varpi_1(t), \varpi_2(t), \ldots, \varpi_Y(t)\}$, which has been discussed in the previous section and can be given in Eq. (25):

$$\mathfrak{s}(t) \triangleq \{D(t), \mathfrak{P}(t), \mathfrak{G}(t), \rho_R^B(t), \varpi_S^B(t)\} \in \mathfrak{S} \tag{25}$$

Due to the continuous state space, the probability can only be expressed in terms of probability density. According to the definition of integral and probability theory, the probability that the system arrives at the state $\mathfrak{s}(t+1)$ under the condition of state $\mathfrak{s}(t)$ and the action $\mathfrak{a}(t) \in \mathfrak{A}$ takes place at the episode t is given by Eq. (26).

$$\mathcal{P}(\mathfrak{s}(t+1)|\mathfrak{s}(t), \mathfrak{a}(t)) = \int f(\mathfrak{s}(t), \mathfrak{a}(t), \mathfrak{s}')\mathrm{d}\mathfrak{s}' \tag{26}$$

4.2 Action Variable and Space

The system requires the agent to decide to store data in the P2P network (on-chain) or in the cloud (off-chain) or to order more storage and handle it afterwards. The set Q can be divided into three parts – Q_1, Q_2, and Q_3 – that represent the three types of actions that data affairs in IoTDs take, in which $Q_1 \cup Q_2 \cup Q_3 = Q$ and $Q_1 \cap Q_2 \cap Q_3 = \varnothing$. The current composite action $\mathfrak{a}(t)$ in the space $\mathfrak{A}$ is denoted by

$$\mathfrak{a}(t) = \{a^{on}(t), a^{off}(t), a^{order}(t)\} \in \mathfrak{A} \tag{27}$$

where $a^{on}(t), a^{off}(t)$, and$a^{order}(t)$ are defined and interpreted as follows:

- $a^{on}(t)$ indicates whether the task will be processed on the blockchain for IoT devices and is defined as a row vector $a^{on}(t) = [a_1^{on}(t), a_2^{on}(t), ..., a_Q^{on}(t)]$ and $a_q^{on}(t) \in \{0, 1\}$, where $a_q^{on}(t) = 0$ means the task will not be processed in the P2P network and $a_q^{on}(t) = 1$ means it will be uploaded to the blockchain (on-chain).
- $a^{off}(t)$ indicates whether the task will be processed in the cloud for IoT devices and is defined as a row vector $a^{off}(t) = [a_1^{off}(t), a_2^{off}(t), ..., a_Q^{off}(t)]$ and $a_q^{off}(t) \in \{0, 1\}$, where $a_q^{off}(t) = 0$ means the task will not be processed in the cloud and $a_q^{off}(t) = 1$ means it will be uploaded to the cloud server (off-chain).
- $a^{order}(t)$ indicates whether there is a need to order more servers in the cloud, which is defined as a row vector $a^{order}(t) = [a_1^{order}(t), a_2^{order}(t), ..., a_Q^{order}(t)]$ and $a_q^{order}(t) \in \{0, 1\}$, where $a_q^{order}(t) = 0$ means that the system does not need to order extra servers for storing tasks and $a_q^{order}(t) = 1$ suggests that it should order some extra servers in the cloud.

4.3 Reward Function

In this proposed on-chain and off-chain data management BIoT system environment, a reward function is formulated to maximise the combination of the reward in the blockchain-driven module and the cloud-driven module and minimise the cost of ordering cloud servers when the storage is insufficient, which involves verification of probability, protocol fees, transaction throughput, comprehensive cloud-based revenue, and the costs of ordering storage.

For the blockchain-based system, the average comprehensive revenue is adopted as the reward of all the SBSs, which can be formulated as tokens in the blockchain. Let R_I^{on} be the income after generating a block, denoted as $R_I^{on} = p^s I^s - (1 - p^s) I^f$, where p^s represents the probability that the SBS succeeds in passing the system's verification, and I^s and I^f refer to tokens of the success income and the failure punishment.

Additionally, SBSs need to consume energy and pay protocol fees in the process of block generation. This can be formulated as $R_C^{on} = C_e^B k_{x,y}^{onc} + C_p^B$, where C_e^B represents the energy cost per Joule and C_p^B is the protocol fee in the blockchain system.

This enables the average comprehensive revenue $R^{on}(t)$ of the on-chain module to be expressed in Eq. (28), where ζ_{on} is the on-chain income factor.

$$R^{on}(t) = \frac{\zeta_{on}}{L_q} M_q (R_I^{on} - R_C^{on}) \wp(t) \tag{28}$$

For the cloud-based system, the average comprehensive revenue $R^{off}(t)$ in the cloud server can be given in Eq. (29), where F_{hz}^C (per HZ), C_E^C(per Joule), and C_s^C(per unit space) represent the fee that IoTDs should pay to infrastructure providers, the energy consumption at the cloud server, and the cost of storage in the memory. Meanwhile, $\zeta_{off}(\zeta_{off} \geq 1)$ is the off-chain income factor. For the off-chain system, the average comprehensive revenue of the cloud server is considered the system reward. It is assumed that the obtained average income is $R^{off}(t)$, as calculated by Eq. (29):

$$R^{off}(t) = L_q (1 - M_q)^{\zeta_{off}} (F_{hz}^C(t) + k_{x,z}^{offc} C_E^C(t) + L_q C_s^C(t)) \tag{29}$$

For server provider, O_p^S represents the order fee per unit time when the storage is inefficient. This enables the total fee for ordering more storage to be expressed by Eq. (30):

$$R^{order}(t) = \lim_{N \to \infty} \sum_{n=1}^{N} \frac{O_p^S}{t^n} \tag{30}$$

By considering all the above rewards, an aggregated reward function for on-chain and off-chain data management can be expressed by Eq. (31):

$$\begin{aligned} R(t) &= \sum_{Q_1} R^{on}(t) + \sum_{Q_2} R^{off}(t) + \sum_{Q_3} R^{order}(t) \\ &= \sum_{q=1}^{Q} a_q^{on}(t) R^{on}(t) + \sum_{q=1}^{Q} a_q^{off}(t) R^{off}(t) \\ &+ \sum_{q=1}^{Q} a_q^{order}(t) R^{order}(t) \\ &= \sum_{q=1}^{Q} (a_q^{on}(t) \frac{\zeta_{on}}{L_q} M_q (R_I^{on} - R_C^{on}) \wp(t) \\ &+ a_q^{off}(t) L_q (1 - M_q)^{\zeta_{off}} (F_{hz}^C(t) + k_{x,z}^{offc} C_E^C(t) \\ &+ L_q C_s^C(t)) + a_q^{order}(t) \lim_{N \to \infty} \sum_{n=1}^{N} \frac{O_p^S}{t^n}) \end{aligned} \tag{31}$$

The cumulative revenue can be formulated by Eq. (32), where γ is the discount factor of the rewards, meaning that the system prioritises early rewards in the process.

$$R^{long} = \max_{\mathfrak{A}} \mathbb{E}\left[\sum_{t=0}^{t=T-1} \gamma R(t) \right] \tag{32}$$

4.4 Customisation of the A3C Algorithm for the Data Management

Deep reinforcement learning (DRL) development has seen researchers consistently propose new algorithms, including algorithms based on value-based and policy-based learning. Combinations of these two methods have produced actor-critic learning, advantage actor-critic learning, and A3C [36]. The A3C algorithm, presented by Google DeepMind in 2016 [36], demonstrates better accuracy and robustness in experiments. As a DRL algorithm, it adopts the deep neural network to take actions based on the system state. Compared with deep Q network (DQN) and Q-learning, A3C can export continuous and discrete actions. Asynchronous actor learners deployed on multiple CPU threads interact, train, and learn simultaneously, as illustrated in Fig. 5. Different agents learn relevant policy in parallel, with parameters mutually independent. This means that more diversified policies can be identified and learned. The pseudo-code for deploying the A3C algorithm for

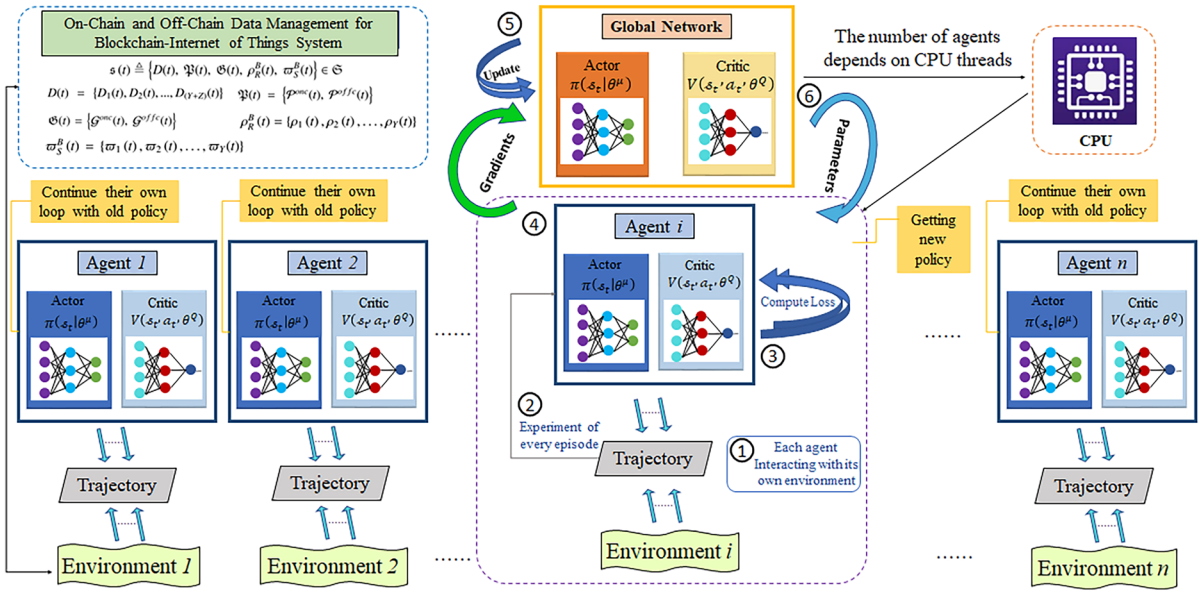

Fig. 5 Algorithm Process of A3C-Based On-Chain and Off-Chain Data Management

on-chain and off-chain data management appears in Algorithm 1.

To be specific, the policy $\pi(\mathfrak{a}_t|\mathfrak{s}_t,\theta)$ describes a series of probabilities in action spaces with the parameter θ in the neural network and the value function $\mathcal{V}(\mathfrak{s}_t;\theta_v)$ represents the excellence degree of a system state with the parameter θ_v in the neural network. Differing from policy gradient methods, A3C can learn and update corresponding parameters of the policy network using the value function – that is, actor and critic. The policy π and the value function $\mathcal{V}$ will be updated simultaneously once the system enters the terminal state or reaches the maximum episode. The discounted returns (the discount factor is $\mathscr{C} \in (0,1]$) are utilised for the policy-based methods, where $\mathfrak{q}$ is the variable and $r_{t+\mathfrak{p}}$ is the immediate reward at episode $t+\mathfrak{p}$, which can be calculated using Eq. (33):

$$R_t(\theta_v) = \mathscr{C}(r) = \mathscr{C}\mathcal{V}(\mathfrak{s}_{t+\mathfrak{q}};\theta_v) + \sum_{\mathfrak{p}=0}^{\mathfrak{q}-1} \mathscr{C} r_{t+\mathfrak{p}} \tag{33}$$

However, avoiding an error value for the action prediction involves using the advantage estimates denoted as $\mathcal{A}(\mathfrak{a}_t,\mathfrak{s}_t) = \mathcal{Q}(\mathfrak{a}_t,\mathfrak{s}_t) - \mathcal{V}(\mathfrak{s}_t)$.

The A3C algorithm adopts deep neural networks to approach these functions due to the calculation difficulties. The formula can be rewritten as follows:

$$\begin{aligned}\mathcal{A}(\mathfrak{a}_t,\mathfrak{s}_t;\theta,\theta_v) &= R_t(\theta_v) - \mathcal{V}(\mathfrak{s}_t;\theta_v) \\ &= \mathscr{C}\mathcal{V}(\mathfrak{s}_{t+\mathfrak{q}};\theta_v) + \textstyle\sum_{\mathfrak{p}=0}^{\mathfrak{q}-1} \mathscr{C} r_{t+\mathfrak{p}} - \mathcal{V}(\mathfrak{s}_t;\theta_v)\end{aligned} \tag{34}$$

For the approaching function, the policy $\pi(\mathfrak{a}_t|\mathfrak{s}_t,\theta))$ utilises the softmax layer due to discrete actions, and the value function can be exported by a linear layer. The algorithm procedure features a central parameter memory that records all the parameters of local agents [37]. At first, agents set parameters according to the centralised brain. Then, they execute formulas, calculate gradients, train neural networks, and send the updates to the central brain. Finally, the brain broadcasts new weights to all networks to share the same parameters, and the process enters the next generation.

The policy loss function is given by Eq. (35), where $\mathfrak{W}(\pi(\mathfrak{s}_t;\theta))$ is the entropy and $\hbar$ is the entropy regularisation term [38]:

$$f_\pi(\theta) = \hbar\mathfrak{W}(\pi(\mathfrak{s}_t;\theta)) + \log\pi(\mathfrak{a}_t|\mathfrak{s}_t,\theta)(R_t - \mathcal{V}(\mathfrak{s}_t;\theta_v)) \tag{35}$$

The gradient of $f_\pi(\theta)$ can be calculated using Eq. (36):

$$\nabla_\theta f_\pi(\theta) = \hbar\nabla_\theta\mathfrak{W}(\pi(\mathfrak{s}_t;\theta)) + \nabla_\theta\log\pi(\mathfrak{a}_t|\mathfrak{s}_t,\theta)(R_t - \mathcal{V}(\mathfrak{s}_t;\theta_v)) \tag{36}$$

Algorithm 1 A3C-Based On-Chain and Off-Chain Data Management Algorithm

Initialisation:

- Global Network: Set the parameters θ for actor network and θ_v for critic network.
- Global Network: Set the parameters θ for actor network and θ_v for critic network.
- Set learning rate.
- Set the number of agents W_{cpu}.
- Set max global episode T_{max}, global counter $T = 0$, and local episode $t = 1$.

Iteration:

1: **while** $T < T_{max}$ **do**
2: **for** $w = 1$ to W **do**
3: Reset parameters in the global network: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
4: Synchronise parameters in the local network: $\theta' = \theta$ and $\theta'_v = \theta_v$.
5: Set $t_{begin} = t$.
6: Observe state $\mathfrak{s}(t)$ from the system.
7: **repeat**
8: Utilise the policy $\pi(\mathfrak{a}(t)|\mathfrak{s}(t), \theta')$ to get the action $\mathfrak{a}(t)$.
9: Take the action $\mathfrak{a}(t)$ and obtain the reward $\mathcal{R}(t)$.
10: Observe the next state $\mathfrak{s}(t+1)$ from the system.
11: $t = t + 1$.
12: $T = T + 1$
13: **until** $t - t_{begin} == t_{max}$ || final state $\mathfrak{s}_f(t)$
14: **if** t mod $t_g == 0$ **then**
15: $R = \mathcal{V}(\mathfrak{s}(t); \theta'_v)$
16: **end if**
17: **for** $t - 1$ to t_{begin} **do**
18: Update Reward: $R = \mathcal{R}(t) + \ell R$.
19: Utilise (36) to compute the policy gradient in the local network: $\nabla_{\theta'} f_\pi(\theta')$.
20: Update the gradient in the global network: $d\theta = d\theta + \nabla_{\theta'} f_\pi(\theta')$.
21: Utilise (38) to compute the value gradient in the local network: $\nabla_{\theta'_v} f_v(\theta'_v)$.
22: Update the gradient in the global network: $d\theta_v = d\theta'_v + \nabla_{\theta'_v} f_v(\theta'_v)$.
23: **end for**
24: Utilise (40) to process asynchronous updates for parameters θ and θ_v in the global network.
25: **end for**
26: **end while**

Consequently, the loss function can be defined by Eq. (37):

$$f_{\mathcal{V}}(\theta_{\mathcal{V}}) = (R_t - \mathcal{V}(\mathfrak{s}_t;\theta_v))^2 \tag{37}$$

Similarly, the gradient of $f_{\mathcal{V}}(\theta_{\mathcal{V}})$ can be calculated by Eq. (38):

$$\nabla_{\theta_{\mathcal{V}}} f_{\mathcal{V}}(\theta_{\mathcal{V}}) = 2(R_t - \mathcal{V}(\mathfrak{s}_t;\theta_v))(-\nabla_{\theta_{\mathcal{V}}} \mathcal{V}(\mathfrak{s}_t;\theta_v)) \tag{38}$$

The DQN sees the Root Mean Squared Propagation algorithm applied extensively to decline the loss function. Here, $\Delta\theta$ and ψ represent the gradient accumulation and the momentum and can be expressed using Eq. (39):

$$g_{net} \leftarrow (1-\psi)\Delta\theta^2 + \psi g_{net} \tag{39}$$

Parameter θ can be updated by the following method, with δ and ϕ representing the infinitesimal positive number and the learning rate:

$$\theta \leftarrow \theta - \phi\Delta\theta(g_{net}+\delta)^{-1/2} \tag{40}$$

5 Simulation Experiments and Analysis

This section analyses the simulation of the proposed DRL approach for on-chain and off-chain data management in BIoT under various parameters to demonstrate the algorithm's performance and practicality. Specifically, TensorFlow 2.11.0 [39] is adopted to construct and validate the novel data management method over the blockchain-IoT network. We use ES9 (ECMAScript 2018) to develop the backbone of the blockchain system, while the core logic of the back end is built on the blockchain with Solidity 0.8.11 in the truffle framework. After introducing the simulation settings, the results are discussed in detail.

5.1 Simulation Settings

Based on the system model and customised A3C algorithm, a series of simulation experiments are conducted to verify the feasibility and performance of data management using the BIoT system environment [40]. Specifically, the BIoT system environment and A3C algorithm are deployed in a GPU server with the specifications shown in Table 3. The other simulation parameters appear in Table 4.

Table 4 Specifications of the Simulation Experiments

Parameters	Description	Values
$\mathcal{D}$	Data size	[0.5 MB, 2 MB]
B_{onc}	Bandwidth	180 kHz
B_{offc}	Bandwidth	500 kHz
p_{max}^{onc}	max power of SBSs	1W
p_{max}^{offc}	max power of cloud	5W
$[I_m, I_M]$	interval of blocks	[1 s, 10 s]
$[L_m, L_M]$	size of blocks	[1 MB, 32 MB]
ξ	average size	300 KB
ϑ_a	actor learning rate	1×10^{-5}
ϑ_c	critic learning rate	2×10^{-4}
ζ_{on}	on-chain factor	50, 150, 200, 250, 300
ζ_{off}	off-chain factor	1, 2, 3, 4, 5

The performance comparison involves comparing the following schemes with the proposed scheme:

- **IDS Scheme** *(Proposed Scheme with Identical Data Size)*: The size of the data input into the system.
- **IDDI Scheme** *(Proposed Scheme with Identical Degree of Data Importance)*: The degree of importance of each data point input into the system.
- **FBSBI Scheme** *(Proposed Scheme with Fixed Block Size and Block Interval)*: The size and frequency of blocks are identical.

5.2 Simulation Results

The simulations of the proposed algorithm primarily illustrate the convergence of the four schemes introduced, where the actor network has the learning rate $\vartheta_a = 2 \times 10^{-5}$ and the critic network has a learning

Table 3 Specifications of the GPU Server

Hardware	Description
Model	SUPERMICRO GPU, SuperServer, SYS-420GP-TNR
Motherboard	X12DPG-OA6, IntelR C621A
Power Supply	4 units of 2000W Redundant Titanium level power supply
CPU	2 units of IntelR XeonR Gold 6338 48 M Cache, 2.00 GHz
RAM	32 units of DDR4-3200 ECC RDIMM 32 GB (Total 1024 GB)
Storage	2 units of 4 TB NVMe PCIe 3.1×4 3D 1DWPD Intel DC P4510
GPU	4 units of NVIDIA Tesla A100 80 GB PCIe

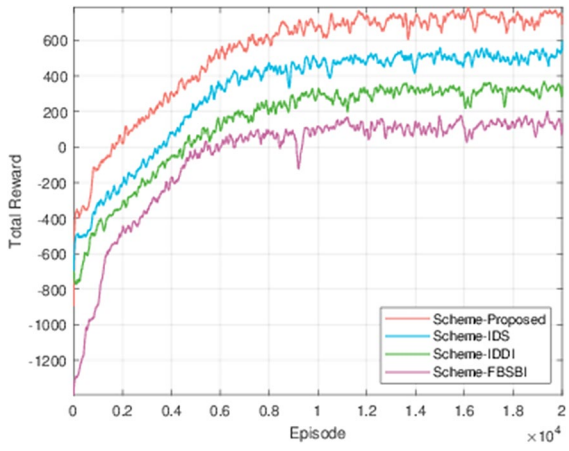

Fig. 6 Total Reward with Schemes

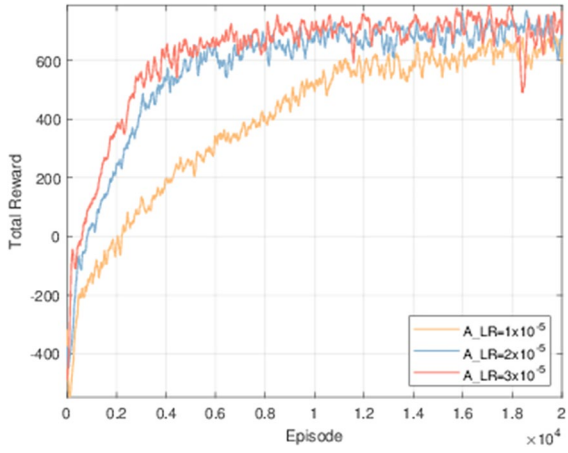

Fig. 7 Total Reward with Different Learning Rate in the Actor Network

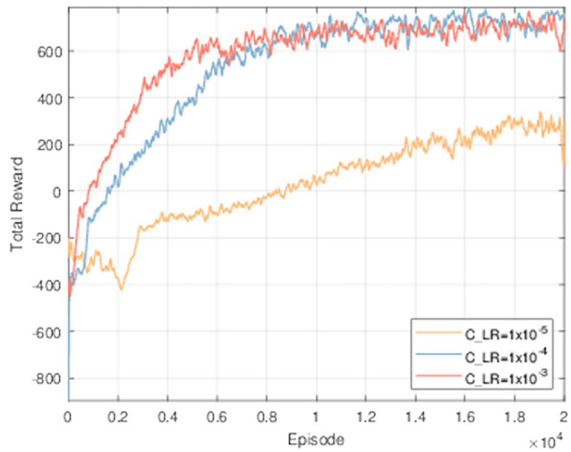

Fig. 8 Total Reward with Different Learning Rate in the Critic Network

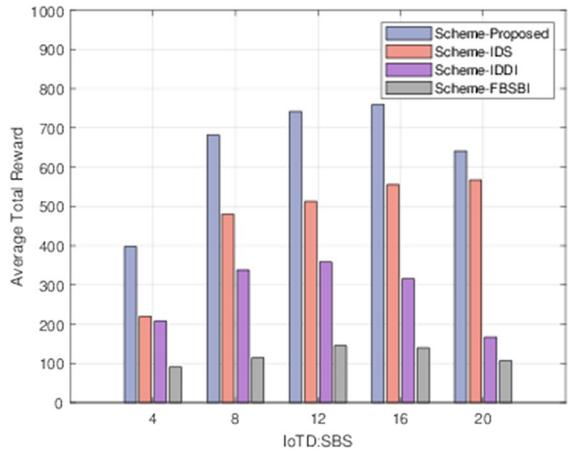

Fig. 9 Average Total Reward with the Ratio of IoTD to SBS

rate $\vartheta_c = 10^{-4}$, as Fig. 6 shows, demonstrating that convergence is excellent before the end of episodes ($T_{max} = 2 \times 10^4$), with the proposed scheme converging most slowly with the maximum reward (around 750) and the FBSBI scheme converging fastest with the minimum reward (around 100).

To discuss the convergence meticulously under different learning rates of two networks in the proposed algorithm, Figs. 7 and 8 show how this factor affects the total reward. In Fig. 7, with the critic learning rate set to the invariable value $\vartheta_c = 10^{-3}$, the total reward has a better rate of convergence under the actor learning rate $\vartheta_a = 10^{-5}$ and $\vartheta_a = 2 \times 10^{-5}$ (after 6×10^3 episodes) compared with $\vartheta_a = 3 \times 10^{-5}$ (after 1.2×10^4 episodes). Similarly, Fig. 8 illustrates that the algorithm first converges with the critic learning rate $\vartheta_c = 10^{-3}$ under the fixed actor learning rate $\vartheta_a = 2 \times 10^{-5}$. However, the convergence demonstrates poor performance when $\vartheta_c = 10^{-5}$.

Figure 9 examines the average total reward with different ratios of IoTD to SBS in the context of comparing the four schemes. As the figure shows, the maximum average reward for the proposed scheme increases to obtain the maximum average reward when the IoTD:SBS = 16. The IDS scheme demonstrates sustainable growth of the average

reward with the increasing of value of IoTD:SBS. The average reward for the IDDI and FBSBI schemes grows and then decreases, reaching its maximum value when the ratio of IoTD to SBS is 12.

Figure 10 shows the average total reward after reaching the convergence with different validation success probabilities in the blockchain in the context of comparing the four schemes. It can be observed that the reward increases as the probability value increases, with all schemes able to obtain the positive reward if the success probability p^s is greater than or equal to 0.6. Meanwhile, all schemes obtain a negative reward when $p^s \leq 0.3$.

Figure 11 illustrates the variation tendency with changing the allowed maximum transmit power on each IoTD. As the figure shows, the average rewards of the four schemes converge at a lower lever when the power is below 1kw. Subsequently, the reward remains relatively unchanged with increased allowable maximum power. This means that excess energy and power may be wasted because the system will not stay under load at all times.

It is also necessary to consider the on-chain, off-chain, and order-storage rewards. Figure 12 compares the total reward under the different income discount factors ζ_{on} and ζ_{off}. When the factor ζ_{on} increases, there is an overall upward trend of the reward that ultimately arrives at the maximum value of $\zeta_{on} = 300$.

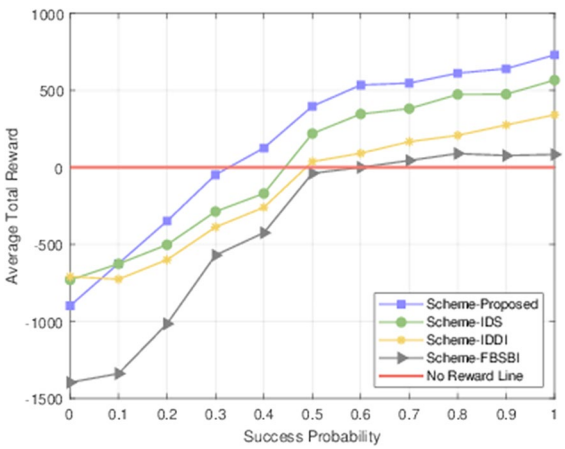

Fig. 10 Average Total Reward with Different Validation Success Probability

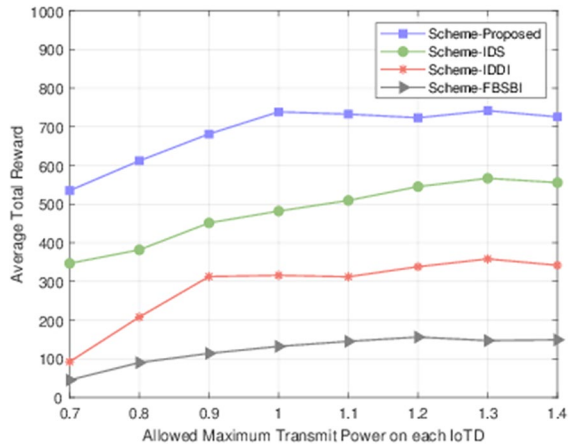

Fig. 11 Average Total Reward with Different Allowed Maximum Transmit Power

For the impact of the factor ζ_{off}, although the reward does not demonstrate particularly obvious variation across the whole process, there is a downward trend, with $\zeta_{off} = 5$.

Figure 13 shows how the income factors (ζ_{on} and ζ_{off}) affect the on-chain and off-chain reward, aligning with the intuition that the ζ_{on} only influences the on-chain system and the ζ_{off} only acts on the off-chain system. The on-chain reward improves with the increase in ζ_{on}, and the off-chain reduces if ζ_{off} grows. This is because the system tends to put the complete data into the blockchain system if the capacity size threshold (ζ_{on}) is relatively large and the complete

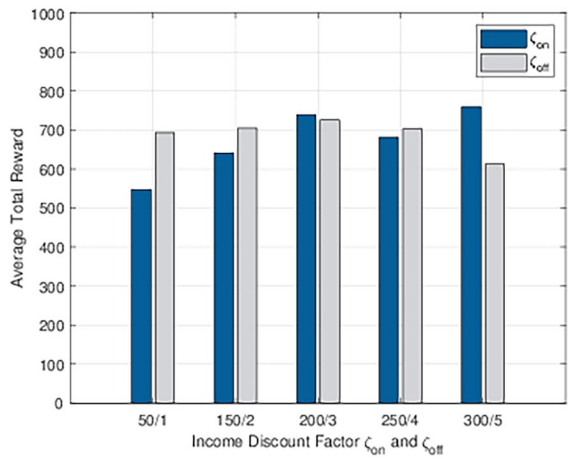

Fig. 12 Average Total Reward with Different Income Factors ζ_{on} and ζ_{off}

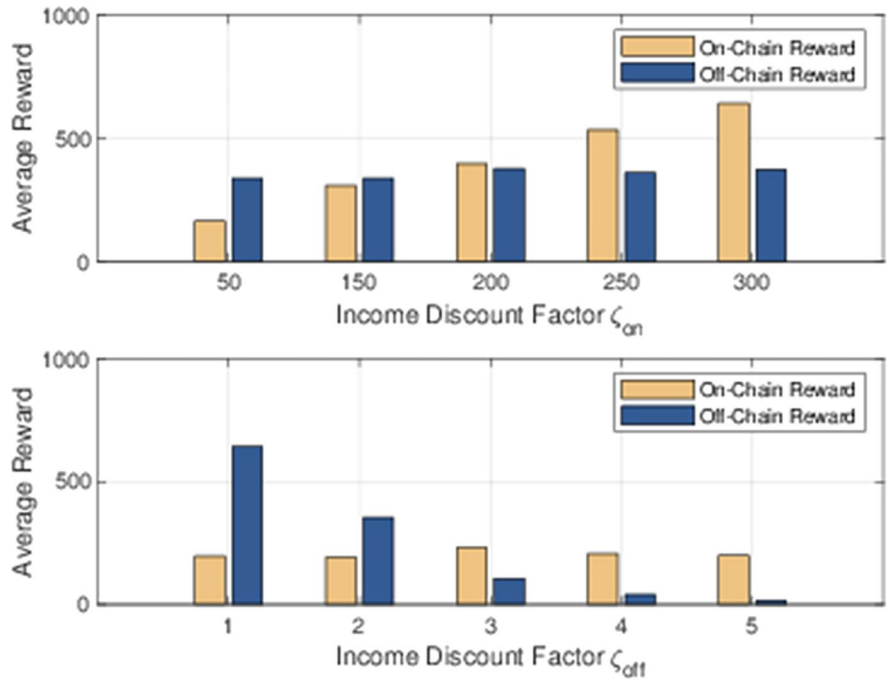

Fig. 13 On-Chain and Off-Chain Average Reward with Different Income Factors ζ_{on} and ζ_{off}

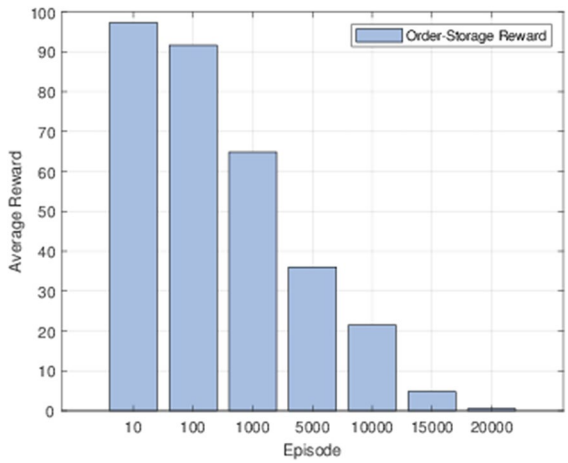

Fig. 14 Order-Storage Average Reward in the Learning Process

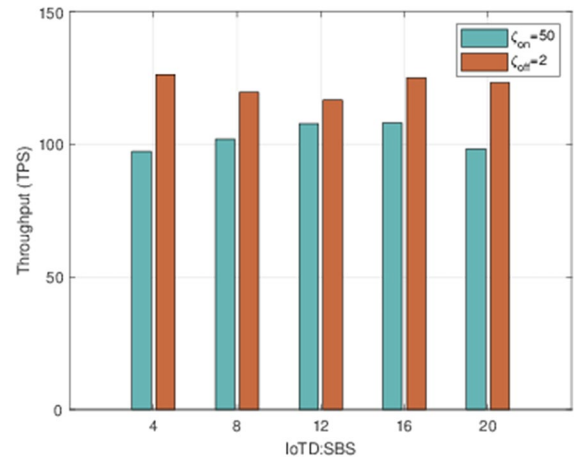

Fig. 15 Throughput Under Different IoTD:SBS

data into a centralised cloud server if the importance factor (ζ_{off}) is small.

Figure 14 illustrates the reward for ordering storage when it is insufficient throughout the process. In earlier episodes, the system obtains greater rewards. However, it obtains almost no reward at the end of the period. This is because, due to the market price of the server providers, the earlier the server is ordered, the cheaper it will be.

Figure 15 shows the throughput of the blockchain module in the proposed system – similar to [41, 42] – and demonstrates that the performance of the blockchain system remains stable. This is because extra computing and power resources balance the blockchain system.

5.3 Practical Implications

Drawing from the preceding simulation results, we can offer some pragmatic insights into the applicability of the proposed mathematical model in practical scenarios, particularly within the realm of blockchain and decentralised environments. The scheme we propose, which utilises the A3C algorithm, proves to be effective and superior for managing off-chain and on-chain data across the blockchain-IoT network. While blockchain-based systems can decentralise on-chain data to enhance cybersecurity and accountability, it is worth noting that the blockchain itself is not inherently designed for large-scale data storage. Consequently, we developed an intelligent agent in a decentralised environment that allocates resources for off-chain and on-chain data management, thereby supporting sustainable blockchain development. As Decentralised Applications (DApps) and Decentralised Autonomous Organisations (DAOs) gain widespread adoption [43, 44], the proposed intelligent agent could be integral, embedded within application systems to provide autonomous data management functionalities. This approach ensures that the scalability and adaptability of application systems are preserved, thereby meeting the complex data management requirements presented by various real-world scenarios.

6 Conclusion

This paper has proposed a BIoT data management system that considers both on-chain (blockchain system) and off-chain (cloud server system) management for IoTDs. Additionally, the system can spend money to order servers from providers when it lacks storage. To formulate the problem accurately, this research has comprehensively integrated the data flow, the transmit power, wireless channel conditions, consensus mechanism, and generation of blocks, modelling these factors as a markov decision process by giving the system state, system action, and total rewards. Subsequently, an A3C algorithm has been used to address the problem by training parameters of neural networks asynchronously. The simulation results show that the proposed scheme performed and converged excellently compared to other schemes under certain restrictions. Notably, different success probabilities in the blockchain system affected the total rewards, and the income factors of on-chain and off-chain modules produced different outcomes. Through the above analysis, the simulation results indicate that the proposed data management approach outperforms other typical data management schemes in terms of convergence and total reward values. It is concluded that the multi-agent deep reinforcement learning approach as an AI method is effective to enhance the scalaiblity, security and reliability of the blockchain-IoT networks.

Having comprehensively reviewed the literature on data management and the Blockchain-enabled Internet of Things (BIoT), we are cognizant of the current research advancements and the unresolved challenges. Future research will delve into the specific modelling and formulation of ordering cloud servers from providers in the BIoT system, which involves ordering varying combinations of CPU cores, GPU cores, and memory. Additionally, we will focus on addressing large-scale IoT devices and data, such as in the supply chain [45, 46]. Our goal is to leverage more sophisticated deep learning methodologies, design efficient data compression and encryption methods, and employ encryption to safeguard data privacy, which can enhance the proposed algorithm's robustness in future BIoT development [47–49]. We also plan to design smart contracts and DAOs to ensure secure access to and use of data. This approach underscores our commitment to advancing the field while ensuring data security and privacy.

Acknowledgements The authors would like to thank the Department of Supply Chain and Information Management & Big Data Intelligence Centre, The Hang Seng University of Hong Kong, and the Department of Industrial and Systems Engineering & Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University for supporting this research study.

Author Contributions Conceptualisation, Y.P.T. and C.K.M.L.; Methodology, Y.P.T. and K.Z.; Formal analysis, Y.P.T. and C.H.W.; Data curation, K.Z. and C.H.W.; Writing—original draft preparation, Y.P.T. and K.Z.; Writing—review and editing, C.H.W., W.H.I., C.K.M.L.; Project administration, C.K.M.L. and W.H.I.; funding acquisition, C.K.M.L. and C.H.W.; All authors reviewed the manuscript before re-submission.

Funding The work described in this paper was partly supported by a grant from Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University (Project

code: CD4E) and a grant from the University Grants Committee of the HK SAR, China (RMGS Project Acc. No.: 700043).

Data Availability The data analysed in this study are available upon reasonable request.

Declarations

Competing interests The authors declare no competing interests.

References

1. Viriyasitavat, W., Da Xu, L., Bi, Z., Pungpapong, V.: Blockchain and internet of things for modern business process in digital economy—the state of the art. IEEE Trans. Comput. Soc. Syst. **6**(6), 1420–1432 (2019)
2. Novo, O.: Blockchain meets iot: an architecture for scalable access management in iot. IEEE Internet Things J. **5**(2), 1184–1195 (2018)
3. Reyna, A., Martn, C., Chen, J., Soler, E., Daz, M.: On blockchain and its integration with iot. challenges and opportunities. Future Gener. Comput. Syst. **88**, 173–190 (2018)
4. Li, Q.K., Lin, H., Tan, X., Du, S.: H∞ consensus for multiagent-based supply chain systems under switching topology and uncertain demands. IEEE Trans. Syst. Man Cybern.: Syst. **50**(12), 4905–4918 (2018)
5. Wang, S., Sheng, H., Zhang, Y., Yang, D., Shen, J., Chen, R.: Blockchain-empowered distributed multi-camera multi-target tracking in edge computing. IEEE Trans. Industr. Inf. **20**(1), 369–379 (2023)
6. Guo, Y., Zhang, C., Wang, C., Jia, X.: Towards public verifiable and forward-privacy encrypted search by using blockchain. IEEE Trans Dependable Secure Comput. **20**(3), 2111–2126 (2022)
7. Ben-Yair: Updating google photos' storage policy to build for the future. Google (2020). https://blog.google/products/photos/storage-changes/. Accessed 30 May 2023
8. Kulshrestha, S., Patel, S.: An efficient host overload detection algorithm for cloud data center based on exponential weighted moving average. Int. J. Commun Syst **34**(4), e4708 (2021)
9. Patel, M., Chaudhary, S., Garg, S.: Machine learning based statistical prediction model for improving performance of live virtual machine migration. J. Eng. **2016**, 3061674 (2016)
10. Bala, A., Chana, I.: Prediction-based proactive load balancing approach through vm migration. Eng. Comput. **32**(4), 581–592 (2016)
11. Saxena, D., Singh, A.K., Buyya, R.: Op-mlb: An online Vm prediction based multi-objective load balancing framework for resource management at cloud datacenter. IEEE Trans. Cloud Comput. **10**(4), 2804–2816 (2021)
12. Zheng, X.R., Lu, Y.: Blockchain technology–recent research and future trend. Enterp. Inf. Syst. **16**(12), 1939895 (2022)
13. Golightly, L., Chang, V., Xu, Q.A., Gao, X., Liu, B.S.: Adoption of cloud computing as innovation in the organisation. Int. J. Eng. Bus. Manag. **14**, 18479790221093990 (2022)
14. Djenouri, Y., Srivastava, G., Belhadi, A., Lin, J.C.W: Intelligent blockchain management for distributed knowledge graphs in IoT 5G environments. Trans. Emerg. Telecommun. Technol. e4332 (2021). https://doi.org/10.1002/ett.4332
15. Djenouri, Y., Yazidi, A., Srivastava, G., Lin, J.C.W.: Blockchain: applications, challenges, and opportunities in consumer electronics. IEEE Consum. Electron. Mag. (2023). https://doi.org/10.1109/MCE.2023.3247911
16. Ma, C.Y., Mo, D.Y.: Integrating internet of things in service parts operations for sustainability. Int. J. Eng. Bus. Manag. **15**, 18479790231165640 (2023)
17. Ni, S., Bai, X., Liang, Y., Pang, Z., Li, L.: Blockchain-based traceability system for supply chain: potentials, gaps, applicability and adoption game. Enterp. Inf. Syst. **16**(12), 2086021 (2022)
18. Marchesi, L., Marchesi, M., Tonelli, R.: Abcdeâ€"agile block chain dapp engineering. Blockchain: Res. Appl. **1**(1–2), 100002 (2020)
19. Song, W., Xiao, Z., Chen, Qi., Luo, H.: Adaptive resource provisioning for the cloud using online bin packing. IEEE Trans. Comput. **63**(11), 2647–2660 (2013)
20. Cao, B., Wang, X., Zhang, W., Song, H., Lv, Z.: A many-objective optimisation model of industrial internet of things based on private blockchain. IEEE Netw. **34**(5), 78–83 (2020)
21. Li, K., Ji, L., Yang, S., Li, H., Liao, X.: Couple-group consensus of cooperative–competitive heterogeneous multiagent systems: a fully distributed event-triggered and pinning control method. IEEE Trans. Cybern. **52**(6), 4907–4915 (2020)
22. Bein, D., Bein, W., Venigella, S.: Cloud storage and online bin packing. In: Brazier, F.M.T., Nieuwenhuis, K., Pavlin, G., Warnier, M., Badica, C. (eds.) Intelligent Distributed Computing V. Studies in Computational Intelligence, vol 382, pp. 63–68. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24013-3_7
23. Mohiuddin, I., Almogren, A., Al Qurishi, M., Hassan, M.M., Al Rassan, I., Fortino, G.: Secure distributed adaptive bin packing algorithm for cloud storage. Future Gener. Comput. Syst. **90**, 307–316 (2019)
24. Woodman, S., Hiden, H., Watson, P.: Applications of provenance in performance prediction and data storage optimisation. Futur. Gener. Comput. Syst. **75**, 299–309 (2017)
25. Ferrer, A.J., Marquès, J.M., Jorba, J.: Towards the decentralised cloud: survey on approaches and challenges for mobile, ad hoc, and edge computing. ACM Comput. Surv. **51**(6), 1–36 (2019)
26. Alam, M.S., Mark, J.W., Shen, X.S.: Relay selection and resource allocation for multi-user cooperative ofdma networks. IEEE Trans. Wirel. Commun. **12**(5), 2193–2205 (2013)
27. Peng, M., Zhang, K., Jiang, J., Wang, J., Wang, W.: Energy-efficient resource assignment and power allocation

in heterogeneous cloud radio access networks. IEEE Trans. Veh. Technol. **64**(11), 5275–5287 (2014)
28. Wang, Y., Sheng, M., Wang, X., Wang, L., Li, J.: Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. IEEE Trans. Commun. **64**(10), 4268–4282 (2016)
29. Burd, T.D., Brodersen, R.W.: Processor design for portable systems. J. VLSI Sign. Process Syst. Signal Image Video Technol. **13**(2), 203–221 (1996)
30. Antshares digital assets for everyone [online]. (2016). Available: https://www.antshares.org. Accessed 30 May 2023
31. Coelho, I.M., Coelho, V.N., Araujo, R.P., Qiang, W.Y., Rhodes, B.D.: Challenges of pbft-inspired consensus for blockchain and enhancements over neo dbft. Future Internet **12**(8), 129 (2020)
32. Liu, M., Yu, F.R., Teng, Y., Leung, V.C.M., Song, M.: Performance optimisation for blockchain-enabled industrial internet of things (iiot) systems: a deep reinforcement learning approach. IEEE Trans. Ind. Inf. **15**(6), 3559–3570 (2019)
33. Clement, A., Wong, E.L., Alvisi, L., Dahlin, M., Marchetti, M.: Making byzantine fault tolerant systems tolerate byzantine faults. In: NSDI, vol 9, pp. 153–168 (2009)
34. Gomaa, H., Messier, G.G., Williamson, C., Davies, R.: Estimating instantaneous cache hit ratio using markov chain analysis. IEEE/ACM Trans. Netw. **21**(5), 1472–1483 (2012)
35. Breslau, L., Cao, P., Li, F., Phillips, G., Shenker, S.: Web caching and zipf-like distributions: evidence and implications. In: IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320), IEEE, vol 1, pp 126–134 (1999)
36. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937. PMLR (2016)
37. Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., Kautz, J.: Reinforcement learning through asynchronous advantage actor-critic on a gpu. arXiv preprint arXiv:1611.06256 (2016)
38. Zhao, S., Gong, M., Liu, T., Huan, Fu., Tao, D.: Domain generalisation via entropy regularisation. Adv. Neural. Inf. Process. Syst. **33**, 16096–16107 (2020)
39. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
40. Belhadi, A., Djenouri, Y., Srivastava, G., Jolfaei, A., Lin, J.C.W.: Privacy reinforcement learning for faults detection in the smart grid. Ad Hoc Netw. **119**, 102541 (2021)
41. Chen, W., Chen, Y., Chen, X., Zheng, Z.: Toward secure data sharing for the iov: a quality-driven incentive mechanism with on-chain and off-chain guarantees. IEEE Internet Things J. **7**(3), 1625–1640 (2019)
42. Feng, J., Yu, F.R., Pei, Q., Chu, X., Du, J., Zhu, L.: Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach. IEEE Internet Things J. **7**(7), 6214–6228 (2019)
43. Liu, A., Zhao, D., Li, T.: A data classification method based on particle swarm optimisation and kernel function extreme learning machine. Enterp. Inf. Syst. **17**(3), 1913764 (2023)
44. Wan, H.C., Chin, K.S.: Exploring internet of healthcare things for establishing an integrated care link system in the healthcare industry. Int. J. Eng. Bus. Manag. **13**, 18479790211019530 (2021)
45. Wang, J.W., Ip, W.H., Muddada, R.R., Huang, J.L., Zhang, W.J.: On Petri net implementation of proactive resilient holistic supply chain networks. Int. J. Adv. Manuf. Technol. **69**, 427–437 (2013)
46. Raj, R., Wang, J.W., Nayak, A., Tiwari, M.K., Han, B., Liu, C.L., Zhang, W.J.: Measuring the resilience of supply chain systems using a survival model. IEEE Syst. J. **9**(2), 377–381 (2014)
47. Guo, P., Hou, W., Guo, L., Sun, W., Liu, C., Bao, H., Duong, L.H.K., Liu, W.: Fault-tolerant routing mechanism in 3d optical network-on-chip based on node reuse. IEEE Trans. Parallel Distrib. Syst. **31**(3), 547–564 (2019)
48. Kakadia, D., Yang, J., Gilgur, A.: Evolved universal terrestrial radio access network (EUTRAN). In: Network Performance and Fault Analytics for LTE Wireless Service Providers, pp. 61–81. Springer, New Delhi (2017). https://doi.org/10.1007/978-81-322-3721-1_3
49. You, C., Huang, K., Chae, H., Kim, B.-H.: Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Trans. Wirel. Commun. **16**(3), 1397–1411 (2016)