A Reputation-enhanced Shard-based Byzantine Fault-tolerant Scheme for Secure Data Sharing in Zero Trust Human Digital Twin Systems

Samuel D. Okegbile, *Member, IEEE,* Jun Cai, *Senior Member, IEEE,* Jiayuan Chen, and Changyan Yi, *Member, IEEE*

***Abstract*—Secure data sharing is imperative in human digital twin (HDT) systems due to the continuous communication requirements among physical and virtual twins, making data security and privacy essential concerns. Previous works have emphasized the significance of blockchain technology in mitigating security challenges within digital twin systems. Nevertheless, existing blockchain-based solutions often fall short of meeting the specific latency and throughput demands of HDT systems, primarily attributed to the complicated consensus process of conventional blockchain solutions. As a result, this paper introduces a novel reputation-enhanced shard-based Byzantine fault-tolerant scheme designed for zero-trust HDT systems. We propose a parallel validation-based reputation-enhanced practical Byzantine fault tolerance consensus framework to address the need for improved throughput and reduced latency during data-sharing processes. This framework incorporates a priority-based block-appending process to prevent forking attacks, ensuring that critical aspects of the blockchain-enabled framework, such as security and decentralization, remain uncompromised. Moreover, we formalize the communication process among validators and their computation resource allocation as a Markov decision process. We then adopt the branching duelling Q-network approach to address the challenge posed by the large dimensions of the action space in our formulated problem. The results demonstrate that the proposed framework significantly enhances authentication, authorization, and validation processes in HDT through increased throughput and reduced latency, providing a robust solution for secure and efficient data sharing in HDT systems.**

***Index Terms*—Blockchain, data sharing, digital twin, parallel validation, zero trust.**

I. Introduction

HUMAN digital twin (HDT) is an emerging technology with the ability to revolutionize the current human-centric environment including the healthcare systems [1]–[3]. When adopted towards enabling personalized healthcare systems, it can provide fast, efficient, and accurate healthcare services following the digital twin (DT) concept by combining various technologies including artificial intelligence, data analytics, internet of things (IoT), and virtual and augmented reality. However, HDT relies on continuous data sharing among physical and virtual twins operating in zero-trust environments where any devices or systems may be compromised, thus the need to ensure that authentication, authorization and validation processes are well managed to facilitate data security and privacy. As a result, blockchain technology has started to gain wide popularity in DT networks to ensure anonymity, authentication, data privacy, trustworthiness, fairness and data integrity [4]–[6].

Generally, blockchain can allow trusts to be established among untrusted parties in a decentralized manner. This decentralized architecture means each node in the blockchain system contains a replica of the cryptographically and tamper-proof chained blocks containing various transaction records as agreed during the consensus process [7]. While blockchain can guarantee secure and privacy-preserving data sharing among untrusted nodes, it suffers from many limitations including high latency, low transactions per second (TPS) rates, and scalability issues. Latency in blockchain-enabled systems can increase significantly with an increase in data size and the number of users/consensus nodes since such will increase the overall processing time due to the complicated validation process. Similarly, scalability issues can arise as the ledger size increases [8]. These have led to various studies on the suitability of blockchain, especially in latency-sensitive services [3], [9]. To facilitate its adoption in HDT, there is a need to redefine the consensus process to ensure that the specific requirements of HDT in terms of latency and throughput are satisfied.

Existing efforts have adopted consensus algorithms such as proof-of-stake (PoS) [10] and practical Byzantine fault tolerance (PBFT) scheme [11], as opposed to proof-of-work (PoW) consensus algorithm, to reduce the latency and improve system performance. Similarly, the TPS scaling method [8] is often adopted by adjusting various blockchain parameters such as block size, block interval, and block producer. A delegated PoS [12] is another consensus protocol that has been proposed to reduce the consensus latency by reducing the number of consensus nodes, although such methods suffer from security and reliability threats. When compared with the proof-based consensus protocol, the BFT-based algorithms provide deterministic execution results, while achieving relatively high performance. This makes such protocols suitable in permissioned blockchains. The BFT-based consensus protocol eliminates the limitations of the proof-based consensus protocol by exchanging data among a group of validators called

S. D. Okegbile and J. Cai are with the Network Intelligence and Innovation Lab (NI^2L), Department of Electrical and Computer Engineering, Concordia University, Montreal QC H3G 1M8, Canada (Emails: samuel.okegbile@concordia.ca; jun.cai@concordia.ca).

J. Chen and C. Yi are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 211106, China (Emails: jiayuan.chen@nuaa.edu.cn; changyan.yi@nuaa.edu.cn).

replicas and can achieve a lower transaction latency than the PoW scheme [13]. The frequent message transmissions among different validators during the consensus process, however, means the communication cost can make such a protocol difficult to apply in HDT networks directly [13]. Thus, a careful performance evaluation is necessary. To date, only a limited number of works have focused on the analysis and management of validation latency in BFT-based blockchain systems owing to its difficulty [14].

Sharding technique can eliminate the scalability issue in PBFT-based blockchain systems by parallelizing transaction processing thereby maximizing the overall throughput in proportion to the number of shards [15]. While such a technique can increase the chance of a single shard takeover – an attack that occurs when compromised nodes take control of the consensus initiative by securing the majority number of validators in a single shard to create a malicious shard – as the number of shards increases [8], its capability to improve the system performance when carefully adopted means the sharding technique is a very useful method. Indeed, the well-discussed trilemma of blockchain systems generally believes that any blockchain framework can only satisfy at most two of the three features: decentralization, security and scalability. Thus, a trade-off relationship often exists among these features since maximizing one feature can degrade the others. Finding an optimal scalability point without compromising security and decentralization is, therefore, essential for applying PBFT-based blockchain in HDT.

This paper thus presents a shard-based blockchain-enabled data sharing (sBeDS) framework for zero-trust HDT systems. To address the issues of scalability, latency and throughput, while ensuring that the decentralization and security features are not compromised, we propose a shard-based reputation-enhanced PBFT consensus framework with a priority-based block appending process to avoid forking attacks [16], which occurs when more than one block points to the same preceding block (often due to the communication latency, where the first generated block is not the first to be appended to the chain), thereby breaking blockchain consistency while degrading the blockchain security. To the best of our knowledge, such an approach has not been considered in any existing study. The contributions of this paper are summarized as follows:

- We introduce a novel framework, called sBeDS, designed to enhance the performance of HDT networks by facilitating multiple concurrent validation processes. This approach aims to improve overall throughput while simultaneously minimizing latency. The shard formation process is redefined as a Shapley value-enhanced transferable utility-based coalitional game, ensuring the generation of high-quality shards. To mitigate the risk of a single-shard takeover, we integrate the concept of trust-based proof of reputation into the PBFT framework.
- We propose a novel approach to block appending, which utilizes a priority-based process informed by prioritized queuing theory to prevent potential forking attacks so as to ensure that only one block is appended at any time.
- We present the sBeDS as a Markov decision process (MDP) to facilitate the optimization of transaction throughput, concurrently minimizing communication and computation latency. Our approach leverages a branching dueling Q-network (BDQ), integrating sharding techniques with deep reinforcement learning (DRL) to enhance overall system performance.

TABLE I
COMMON NOTATIONS USED

Notation	Definition
B	Maximum block capacity limit
t_{int}	Time interval
a_{tran}	Number of arrived transactions during any t_{int}
$P_{a,b}$	Offloading power of node a when offloading to b
$h_{a,b}$	Channel gain between any nodes a and b
W; σ^2	Bandwidth; Noise signal power
χ; S^B	Average transaction size; Block size
N	Total number of validators
$N_{s,k}$	Number of validators in shard k
K	Number of shards
f	Total number of possible faulty/malicious validators
f_k	Number of possible faulty validators in shard k
$con_{v_i^k,v_j^k}$	Number of consistent responses of validator v_j^k at v_i^k
$incon_{v_i^k,v_j^k}$	Number of inconsistent responses of v_j^k at v_i^k
d_{th}; $R_{d_{th}}$	Pre-defined reputation threshold; Data rate threshold

The remainder of this paper is organized as follows. In Section II, we discuss various related studies. The details of the proposed sBeDS framework are presented in Section III. Section IV introduces the details of the reputation-enabled PBFT consensus protocol, while Section V presents the analysis of the relevant metrics of interest. In Section VI, the corresponding resource optimization problem is formulated and Section VII shows the simulation results. Finally, Section VIII concludes the paper. Common notations used in this paper are presented in Table I.

II. RELATED WORK

In this section, we review previous works on the performance issues in blockchain, reputation-enhanced consensus schemes, performance optimization techniques and parallel validation methods in blockchain systems.

A. Blockchain-based zero-trust system performance issues

Blockchain plays a crucial role in addressing security concerns within zero-trust environments. In [17], it was implemented to ensure anonymity, fairness, etc. in zero-trust IoT environments. The work in [18] utilized a sharding blockchain in a zero-trust cloud-edge-end environment for enhanced performance. Efforts to optimize system performance in blockchain-enabled data-sharing frameworks were explored in [19], [20]. Addressing consensus latency, joint modeling of transmission and consensus latency was also conducted in [21]. Generally, two types of consensus protocols exist: proof-based and BFT-based, with BFT-based consensus, preferred in large-scale systems for its superior performance [14].

The PBFT consensus protocol found application in various domains such as internet of vehicle (IoV) networks [19], industrial IoT systems [20], and IoT [22]. Additionally, the

authors in [23] presented a BFT decentralized federated learning method for autonomous vehicles with privacy preservation. Evaluation of BFT fault-tolerance under different network settings, considering throughput and latency, was conducted in [24]. PBFT performance was enhanced in [25] through an Eigen trust-based approach, ensuring the selection of high-quality nodes for consensus groups. Performance modeling of BFT schemes aimed at minimizing consensus latency was explored in [26], [27] through a multi-block approach [26] and multi-core processors [27]. Notably, none of the works [17], [19]–[27] considered parallel validation, with only [19], [20], [22] addressing both validation and message exchange latency in their analyses.

B. Reputation-enhanced consensus schemes and performance optimization techniques

Consensus mechanisms are vital in blockchain systems. Any set of nodes selected to participate in the validation process can influence the performance of such systems since this set of distrusting nodes must reach a consensus to append any block to the blockchain. Trust-based analyses, as observed in [25], enhance performance in blockchain-enabled data sharing. Integrating trust-centric schemes is thus essential as demonstrated in a trust evaluation mechanism (ATEM) for node trustworthiness [16] and in an optimized PBFT (T-PBFT) for optimized consensus [25]. Furthermore, a trust-enabled blockchain (TeB) framework was proposed in [28]. Similarly, a reputation-based voting scheme (BIoV) was also presented in [29], while an attack-resistant trust model based on multidimensional trust metrics (ARTMM) was proposed in [30] to reduce unreliable underwater communications. However, existing schemes often neglect link quality when estimating node reputation or trust, a gap addressed in the proposed sBeDS framework as presented in Table II.

To further enhance system performance in blockchain-enabled data sharing, recent efforts have leveraged optimization techniques like DRL. These techniques efficiently optimize resources in complex networks, as demonstrated in a DRL-based framework for blockchain-enabled IoV presented in [19]. The framework focused on maximizing transaction throughput in an IoV setting, addressing challenges posed by unstable network connections. Recognizing the potential security and scalability issues associated with data sharing in such environments, the authors in [31] integrated multi-access edge computing (MEC) and blockchain technologies in autonomous vehicles. This integration aimed to optimize transaction throughput and reduce latency in the MEC system. The joint optimization problem was formulated as a MDP using DRL. In similar works, an advantage actor-critic algorithm was used to solve the DRL-enabled optimization problem in [32]. A new BDQ approach was proposed in [33] to enable the use of discrete-action algorithms in DRL for high-dimensional discrete or continuous action space domains. Furthermore, the performance of user sharing-based caching was improved in [34] through a blockchain-incentivized device-to-device and MEC caching system. The weighted sum of the computation rate and the transaction throughput was maximized in [28] by jointly optimizing the cooperative offloading decision and resource allocation. While DRL-based optimization techniques exhibit promise, the reliance on a single validation process limits their overall performance in blockchain-enabled data-sharing systems.

C. Parallel validation methods

Parallel validation methods can significantly improve the validation process thereby improving the overall system performance [18], [35], [36]. A parallel blockchain validation was first considered using the sharding technique in [8]. In [35], shards were created by considering shard trust difference, communication delay difference and node count difference among shards. A clustering-based sharded blockchain strategy for collaborative computing in IoT networks was similarly presented in [37], while the work in [36] analyzed the security issues in sharding blockchain-based fog computing networks. Sharding technique was also adopted in [38]–[40] where a dynamic blockchain sharding scheme based on the hidden Markov model was formulated in [39] while the work in [40] employed a sharding scheme in edge computing architecture. Under the sharding method, the blockchain validators are clustered into a different group of shards such that each shard independently creates and validates blocks through intra-shard consensus processes. In [8], [36], [37], each validated block from each shard was merged and validated again by a final consensus process (following a double-layer consensus mechanism) before the new block was appended to the chain.

Although increasing the number of shards to increase the TPS can compromise security, sharding techniques have proven effective in enhancing blockchain scalability and throughput [8]. To address security concerns, existing shard-based techniques typically employ a double-layer consensus mechanism, which may increase consensus latency. In this paper, we propose an integration of PBFT and trust-based proof-of-reputation consensus mechanisms. A shard is only created if security constraints are met, maintaining security levels as in the conventional PBFT schemes while concurrently improving scalability, decentralization, and throughput through parallel validation. An evaluation of such a system becomes imperative, emphasizing the need to assess its overall performance and effectiveness.

Therefore, we carry out a performance analysis and optimization of such a parallel validation-based reputation-enhanced PBFT consensus framework to provide valuable solutions for the development of HDT.

III. System model

This section presents the general framework of the sBeDS framework. We define a shard as each partition (i.e., cluster) in the blockchain system containing a group of validators with the ability to make distinctive and independent validation decisions compared to other groups.

A. Network model

Zero-trust HDT framework aims to combine elements from various cybersecurity and identity management principles to

TABLE II
COMPARISON WITH EXISTING REPUTATION-BASED SCHEMES

Reputation-based Schemes	T-PBFT	ATEM	TeB	BIoV	ARTMM	sBeDS
Historical Reputation	✓	✓	✓	✓	✓	✓
Direct Trust	✓	X	✓	✓	✓	✓
Indirect Trust	✓	X	✓	✓	✓	✓
Recommendation Reliability	X	✓	✓	X	✓	✓
Consensus Protocol	BFT-based	Proof-based	BFT-based	Proof-based	–	BFT-based
Considered Link Quality	X	X	✓	X	✓	✓
Considered relationship between Reputation and f	X	X	X	X	X	✓

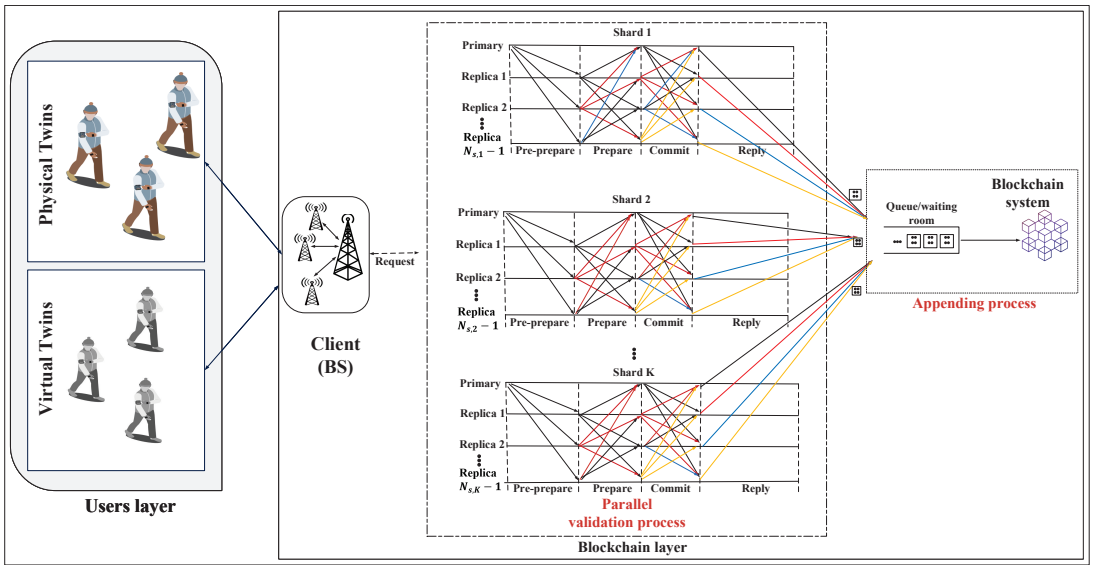

Fig. 1. sBeDS framework.

enhance security and privacy in the HDT. Since HDT involves collaborations from many untrusted users in both physical and virtual environments, verification and authentication become mandatory to minimize the attack surface and prevent unauthorized access. Hence, we introduce a sBeDS-enabled zero-trust HDT system, where users including physical and virtual twins can be data owners (DOs) or data requesters (DRs) as shown in Fig. 1. The validation process is parallelized such that $K \geq 1$ number of validation processes take place simultaneously, where only one block is validated at any time in each shard. We consider the time to be discrete, where time is segmented into equal time slots $t = 1, ..., \infty$. Thus, arrivals and departures (i.e., completion of validation processes) of blocks occur within the time slot boundaries, i.e., the departure from the system can only occur in the interval (t^-, t), while the arrival can only occur in the interval (t, t^+).

Similar to [31], [32], we consider orthogonal spectrum for the transmission between users $U = \{u_1, u_2, ..., u_N\}$ and the group of base stations (BS), called clients, where users offload transactions to the blockchain layer for validation. Each block is generated immediately after the maximum block capacity limit B is reached. When the block maximum capacity is not reached within any time slot interval t_{int}, all arrived transactions at the client during the interval t_{int} are packaged into a single block. Thus, each block is made up of at least one transaction.

After the generation of any block, such a generated block is forwarded to one idle shard $k \in \{1, 2, ..., K\}$. We considered the entire system to be stable, hence the block generation rate is less than the average joint validation rate. With this, there is always at least a shard available for each generated block. We adopt the PBFT consensus protocol. Thus, each block goes through each stage of the PBFT protocol. It becomes immediately clear that such a system may suffer from forking attacks, for instance, when the first arriving block for the appending process is not the first to be generated among the blocks currently under validation. To address this issue, we propose a priority-based block appending process, where shards are classified into different classes of priority based on their reputation scores (obtained from the average reputation scores of validators within each shard) and the number of currently active shards. That is, any idle shard with the highest reputation score takes the next priority after the currently active shards. Hence, any newly generated block is forwarded to the idle shard with the highest priority such that the block from the shard with the highest priority is appended first to the chain. More details about the shard-based validation process will be provided in the next sections.

Consider a tagged user u_0 and its associated client, located at the origin 0. The achievable data rate between the user u_0 and the associated client [41], [42] can be captured as

$$R_{u_0,0} = W \log_2 \left(1 + \frac{P_{u_{0,0}} h_{0,0}}{\sigma^2}\right), \quad (1)$$

where W is the bandwidth. Given that χ is the average size of each transaction, its average offloading time is given as

$$T_{u_i,0}^{\text{off}} = \frac{\chi}{R_{u_i,0}}. \tag{2}$$

Considering any block generation phase in a given slot, let the first transaction arrive at the client at $t = t_0$ and let the Bth transaction arrive at $t = t_B$. The time to generate a block (TGB) is obtained as

$$TGB = \min\{t_B - t_0, t_{int}\}. \tag{3}$$

B. Shard formation

The shard formation is an essential part of the validation process and can be very complicated due to the need to maintain a high level of security in each shard. Since a high data rate is central to improved performance in PBFT-based systems, we formulate the shard formation process as a Shapley value-enhanced transferable utility-based coalitional game and incorporate a trust-based reputation framework to ensure security in each shard. This ensures only validators within the same coverage area [43] can form a shard subject to PBFT consensus protocol constraints.

A transferable utility-based coalitional game is formulated as a pair (V, v), where $V = \{v_1, v_2, ..., v_N\}$ is the set of validators (i.e., players) and $v : 2^N \rightarrow \mathbb{R}$ is a mapping with $v(\phi) = 0$. The mapping v is generally known as the value function, such that for any subset V_k of V, $v(V_k)$ represents the value of the coalition V_k and captures the overall transferable utility that is achievable by validators in V_k without the contribution of validators $V \backslash V_k$. We know that the set of validators V is the grand coalition with $v(V)$ representing its value. Define a function $F_d : X \times X \rightarrow \mathcal{R}^+ \cup \{0\}$, where $F_d(v_i, v_j), \forall v_i, v_j \in V$ indicates the distance between v_i and v_j while $F_d(v_i, v_i) = 0$. If $\bar{f} : \mathcal{R}^+ \cup \{0\} \rightarrow [0, 1)$ is a monotonically nondecreasing dissimilarity function over F_d given that $\bar{f}(0) = 0$ and $f : \mathcal{R}^+ \cup \{0\} \rightarrow [0, 1)$ represents the corresponding similarity given that $f(.) = 1 - \bar{f}(.)$, then the shard formation approach can be seen as either grouping together validators with less dissimilarity as given by $\bar{f}$ or equivalently validators with more similarity as in f.

At the beginning of every shard formation process, validators $v_i \in V$ with reputation values below a pre-defined acceptable threshold a_{th} are removed from the blockchain system and are not allowed to participate in the shard formation process. Similarly, any newly joined validator is assigned the reputation value of a_{th}. Every eligible validator then interacts with other validators with the aim of maximizing its value (i.e., gain). Let $v(\{v_i\}) = 0, \forall v_i$ such that v_i does not belong to any shard. In addition, given any shard V_k,

$$v(V_k) = \frac{1}{2} \sum_{v_i, v_j \in V; v_i \neq v_j} f(F_d(v_i, v_j)). \tag{4}$$

This approach captures the overall value of any shard as the sum of pairwise similarities between the validators since points within a cluster are generally similar to each other. For any two validators $v_i, v_j \forall F_d(v_i, v_j) \leq \epsilon$ in the convex game setting where $\epsilon \rightarrow 0$, their Shapley values are almost equal [44]. From this, the Shapley value-enhanced shard formation is realized through the transferable utility-based coalitional game following Algorithm 1. Note that Algorithm 1 relies on the use of the similarity threshold parameter s_{tr} to assign validators with almost equal Shapley values to the same shard while the validators selected as primary validators (i.e., the center of each shard) are reasonably far apart.

Let $N \geq 3f + 1$ represent the total number of validators available during any shard formation process, where f denotes the total number of possible faulty or malicious validators, while $N_{s,k} \leq N, (\forall k \in \{1, 2, ..., K\}$ and $N \geq 3f + 1)$, captures the number of validators in shard k. It is worth noting that, an increase in $N_{s,k}$ increases the security level since a higher number of possibly malicious validators within shard k, $f_k \geq \frac{N_{s,k}-1}{3}$, will be required to compromise such a shard. On the other hand, a lower $N_{s,k}$ increases the total number of available shard K thereby improving the system decentralization and scalability levels at the expense of security. In addition, both communication overhead and latency within each shard increase with $N_{s,k}$ since a larger amount of messages will be required to reach a consensus, noting that the time consumption increases exponentially with $N_{s,k}$. It becomes immediately clear that if we reduce the number of possible faulty or malicious validators f_k within each shard k, we can minimize $N_{s,k}$, such that the security level remains acceptable (as in the single shard PBFT-based consensus scheme), while the decentralization, throughput and scalability levels increase. Thus, given N, it is desirable to

$$\min_{v_i \in V} f \leq \frac{N-1}{3}, \tag{5}$$

$$\text{s.t.} \quad f_k \leq \frac{N_{s,k}-1}{3}, \forall k \in \{1, 2, ..., K\}, \tag{5a}$$

$$\sum_k N_{s,k} \leq N, \tag{5b}$$

$$\sum_k f_k \leq f. \tag{5c}$$

The constraints in (5a) – (5c) ensure that honest consensus is always guaranteed for each shard by enforcing a higher value of a_{th}. In addition, each validator $v_i \in V$ is required to solve a cryptographic puzzle during its registration on the system to obtain its unique identity $< Id_{v_i}, D_{v_i}^{\text{trust}} >$ such that the cost of whitewashing outweighs its benefit.

C. PBFT consensus protocol

The PBFT consensus process generally has five phases as shown in Fig. 1: REQUEST, PRE-PREPARE, PREPARE, COMMIT, and REPLY. During the REQUEST phase, the client forwards any newly generated block to the selected shard for validation. The primary validator of the selected shard then verifies the message authentication code (MAC) of each transaction in the received data block during the PRE-PREPARE phase. After the initial verification, the primary broadcasts the block to $N_{s,k} - 1$ replicas for validation.

In the PREPARE phase, each replica authenticates the received pre-prepare decision message and exchanges MACs with all other replicas within the corresponding shard to

Algorithm 1: Shapley value-enhanced shard formation process

Input: Set of validators $V = \{v_1, v_2, ..., v_N\}$; similarity threshold parameter $s_{tr} \in (0, 1]$
Output: Set of shards
For i to n **do**
    Compute the Shapley value of each validator using $\phi_i = \frac{1}{2} \sum_{v_j \in V; i \neq j} f(F_d(v_i, v_j))$.
End For
Initialize $\mathcal{Q} = V$; $\mathcal{K} = \{\}$
While $\mathcal{Q} \neq \{\}$
    $p = \arg\max_{i:v_i \in \mathcal{Q}} \phi_i$
    $\mathcal{K} = \mathcal{K} \cup \{v_p\}$
    $P_p = \{v_i \in \mathcal{Q} : f(F_d(v_p, v_j)) \geq s_{tr}\}$
    $\mathcal{Q} = \mathcal{Q} \backslash P_p$
Apply k-means algorithm using $\mathcal{K}$ as the shard centers (primary validators) subject to PBFT constraints.

ensure that the same block was received from the primary. The validators then enter the COMMIT phase, where the block is validated. After validation, each validator sends its validation outcome to the client during the REPLY phase, where the block from each shard is appended to the chain if the consensus is reached among validators in such a shard and subject to the inter-shard priority class. Generally, a block moves from one phase to the next phase of the PBFT scheme if two-thirds of the responses from the participating nodes consent [8].

D. Reputation model

We adopt a trust-based reputation model where each validator $v_i^k \in V_k$ generates a reputation opinion or value about each validating pair $v_i^k, v_j^k \in V_k, \forall j \neq i$ within the same shard k after every validation process. If the received validation decision from v_j^k is consistent with the majority of the received decisions, the validator v_i^k updates its direct consistent value $con_{v_i^k, v_j^k}$ of validator v_j^k, otherwise, it updates its direct inconsistent reputation value, $incon_{v_i^k, v_j^k}$. These values are continually aggregated and securely stored in the blockchain system and are used during the replica selection process. The reputation value of each validator is calculated using both direct and indirect trust values as will be discussed in Section IV. Generally, the direct trust value of any validator v_i^k for another validator v_j^k is defined as trust values obtained through previous direct transactions between the pair v_i^k and v_j^k, while indirect trust values of validator v_i^k for any validator v_l^k is based on the recommendation of another validator (for instance v_j^k) based on the previous transactions between v_j^k and v_l^k.

Note that the reputation score of each validator is obtained through the accumulation of its previous transactions. Hence, the validator with a high reputation score has a high behaviour consistency and thus high trustworthiness as in [16], [25], [28]–[30]. With this, we know that f_k depends on the reputations of selected validators within any shard k. It is worth mentioning that, these reputation opinions are not only based on the actual intentions of the participating validators but are also influenced by link quality [30]. As a result, a clustering-based shard formation technique is adopted, such that the distance between nodes within the same shard is limited, thus reducing the effect of bad link quality. To prevent misrepresentation of trust values (e.g., v_i^k generating a wrong recommendation of v_j^k to mislead other validators), we integrate recommendation reliability into the sBeDS framework to ensure validators with inconsistent recommendations are always detected and penalized.

E. Association rule model

We adopted a tuple $G(S, B_l, V, K)$ to describe the presented shard-based blockchain-enabled data-sharing framework, where S is the set of transactions and B_l is the set of blocks. We can use the weight matrix $SB = [sb_{ij}]$ to represent the association relations between transactions and blocks, where $sb_{ij} = 1$ indicates that a transaction s_i is packaged into a block b_j and $sb_{ij} = 0$ if otherwise. Thus, a transaction can only be packaged into one block, such that $\sum_j sb_{ij} = 1$. The transaction-block association matrix is generally of the form

$$\begin{bmatrix} sb_{11} & sb_{12} & \dots & sb_{1B} \\ sb_{21} & sb_{22} & \dots & sb_{2B} \\ \vdots & \vdots & \dots & \vdots \\ sb_{S1} & sb_{S2} & \dots & sb_{SB} \end{bmatrix}. \tag{6}$$

Similarly, a block b_i can only be validated in a single shard k_j, such that the weight matrix $BK = [bk_{ij}]$, with $\sum_j bk_{ij} = 1$, while a validator v_i can only belong to one shard k_j at any observation time, with the weight matrix $VK = [vk_{ij}]$ and $\sum_j vk_{ij} = 1$. Thus, the block-shard and validator-shard association matrices follow the same form as in (6).

One crucial consideration in the shard-based blockchain is cross-shard transaction processing. This includes transactions that require more than one shard for processing, making the need to minimize the cross-shard validation overhead essential. By adopting eventual atomicity [45], each transaction that requires cross-shard processing can be split into multiple atomic transactions by the client. Each of these atomic transactions relates to different shards and is processed in parallel at different shards.

The proposed sBeDS scheme guarantees the ACID property – atomicity, consistency, isolation, and durability – by ensuring (i) atomicity: a transaction is only sent for the block appending process if it has been successfully validated through the five stages of PBFT; (ii) consistency: each block of transactions remains unchanged throughout the validation process; (iii) isolation: the conditions $\sum_j sb_{ij} = 1$, $\sum_j bk_{ij}$ and $\sum_j vk_{ij}$ are always true; and (iv) durability: blockchain is immutable, thus each validated block is irreversible.

F. Attack vectors and threat model

We consider a Byzantine adversary model in which any validator is capable of engaging in arbitrary and malicious behaviour, including sending conflicting messages or colluding

with other malicious entities, leveraging their full knowledge of the system. Such a validator can exhibit various forms of arbitrary behaviour, such as sending incorrect validation decisions, providing inconsistent information, or refusing to cooperate. Byzantine validators may collaborate to maximize the impact of their attacks, aiming to disrupt consensus protocols, compromise data integrity, or cause system failure. Generally, BFT protocols are designed to withstand attacks from Byzantine adversaries by integrating redundancy, cryptographic techniques, and consensus mechanisms that can withstand a certain proportion of malicious behaviour while maintaining system integrity and correctness.

Notwithstanding this, any PBFT-based solution may still be susceptible to network threats, such as Sybil Attacks, where an attacker creates multiple fake validating nodes to gain control or influence over the consensus process. In such scenarios, the attacker could manipulate the voting process, disrupt consensus, or compromise the integrity of the validation system. Another potential attack vector is denial-of-service attacks, wherein the validation network is overwhelmed with malicious requests, disrupting communication and prolonging the consensus time.

The incorporation of trust-based reputation systems in our proposed solution ensures effective control of these vulnerabilities. The proposed framework can tolerate a certain number of Byzantine nodes without compromising the integrity of the consensus, and it is capable of identifying and excluding Byzantine nodes from the consensus process. The multi-stage validation process of the PBFT protocol, coupled with the adopted cryptographic-enabled registration process, enhances communication integrity and node authentication.

IV. Reputation-enabled PBFT consensus framework

In this section, we present the details of the adopted reputation-enabled PBFT consensus algorithm focusing on the analysis of direct and indirect trust values. This will be integrated into the analyses in Section V.

A. Integrated trust-based reputation and PBFT scheme

Given $v_i \in V$, the number of validators in each shard $N_{s,k}$ from (5) satisfies the PBFT constraints

$$N_{s,k} \geq 3f_k + 1, \forall \sum_k N_{s,k} \leq N, k \in \{1, 2, ..., K\}. \quad (7)$$

As mentioned in Section III, we integrate a trust-based reputation scheme with the PBFT-based validation process to minimize the number of malicious validators. Given a predefined reputation threshold $d_{th} > a_{th}$, each validator v_i^k with reputation values $D^{\text{trust}}_{B,v_i^k} < d_{th}$, as evaluated by the blockchain system, is tagged as a node with a high probability of failure. The blockchain system continuously compares the reporting trust values of each node and removes nodes with low reputation values to minimize the percentage of nodes at a heightened risk of failure within the network. The updated aggregated trust values of validators $D^{\text{trust}}_{B,v_i^k}$ are stored in the blockchain to facilitate the removal of validators with low reputation values. With this, the blockchain system can estimate the reliability of received indirect trust values (i.e., recommendations) from each node and penalize nodes with malicious recommendations.

During each stage of the PBFT, each tagged validator $v_j^k \in V_k$ develops a direct trust value for every other validator $v_i^k \in V_k, (\forall j \neq i, V_k \in V)$ within the same shard k. These trust values are forwarded to the trust aggregation server located in the blockchain system at the end of each observation phase via dedicated error-free communication links. Thus, the blockchain system maintains a continuously updated and aggregated indirect trust value for each validator $v_i \in V$ based on the direct trust values received from other validating nodes. To capture the possible influence of bad link quality, the PBFT constraints in (5) should be achieved as a function of the reputation value and the achievable data rate of each validator within the coverage region of the selected primary. Hence, we can allow the transmission data rate R_{0,v_i^k} between the center of any shard k and each validator $v_i^k \in V_k, \forall p \neq i$ to depend on the overall reputation value $D^{\text{trust}}_{v_p,v_i}$ of each validator $v_i^k \in V_k$. With this, any validator v_i^k at time t is selected to join a shard k based on Algorithm 1 if the $R_{0,v_i^k}(t) \geq R_{d_{th}}$.

Between any two nodes v_i^k and v_j^k, let $D^{\text{trust}}_{v_i^k,v_j^k} \in [0, 1]$ denote the trust value of the node v_j^k from node v_i^k. The data transmission rate of node v_j^k received at the node v_i^k can be obtained from (1) as

$$R_{v_i^k,v_j^k}(t) = WD^{\text{trust}}_{v_i^k,v_j^k}(t) \log_2\left(1 + \frac{P_{v_i^k,v_j^k} h_{v_i^k,v_j^k}}{\sigma^2 + \sum_{v_l^k \in V_k \setminus v_j^k} P_{v_i^k,v_l^k} h_{v_i^k,v_l^k}}\right). \quad (8)$$

From (8), the transmission time of any block of size S^B between any two validators $v_i^k, v_j^k \in V_k$ is thus given as

$$\varphi_{v_i^k,v_j^k} = \frac{S^B}{R_{v_i^k,v_j^k}}, \forall R_{v_i^k,v_j^k} \neq R_{v_j^k,v_i^k}. \quad (9)$$

Next, we present the analysis for direct and indirect trusts between any two nodes, which helps to obtain $D^{\text{trust}}_{v_i^k,v_j^k}$.

B. Direct trust

The direct trust between any two validators v_i^k and v_j^k is obtained following the subjective logic framework, described as a tuple $\omega_{v_i^k,v_j^k} = \{b_{v_i^k,v_j^k}, d_{v_i^k,v_j^k}, v_{v_i^k,v_j^k}\}$, where $b_{v_i^k,v_j^k}$ and $d_{v_i^k,v_j^k}$ are the belief and disbelief respectively of node v_i^k for node v_j^k, while $v_{v_i^k,v_j^k}$ is the degree of uncertainty in the belief system. These parameters satisfy the constraints

$$b_{v_i^k,v_j^k}, d_{v_i^k,v_j^k}, v_{v_i^k,v_j^k} \in [0, 1], \quad b_{v_i^k,v_j^k} + d_{v_i^k,v_j^k} + v_{v_i^k,v_j^k} = 1. \quad (10)$$

Hence, the reliability level of a validator v_j^k from any validator v_i^k can be obtained as

$$RD_{v_i^k,v_j^k} = b_{v_i^k,v_j^k} + \varepsilon v_{v_i^k,v_j^k}, \quad (11)$$

where $0 \leq \varepsilon \leq 1$ captures the influence of the trust uncertainty. From Section III-D, we can define

$$\begin{aligned} b_{v_i^k,v_j^k} &= \frac{con_{v_i^k,v_j^k}(1-v_{v_i^k,v_j^k})}{con_{v_i^k,v_j^k}+incon_{v_i^k,v_j^k}}, \\ d_{v_i^k,v_j^k} &= \frac{incon_{v_i^k,v_j^k}(1-v_{v_i^k,v_j^k})}{con_{v_i^k,v_j^k}+incon_{v_i^k,v_j^k}}, \\ v_{v_i^k,v_j^k} &= 1-Q_{v_i^k,v_j^k}. \end{aligned} \tag{12}$$

Note that $con_{v_i^k,v_j^k}$ and $incon_{v_i^k,v_j^k}$ represent the overall historical number of consistent and inconsistent responses received from any v_j^k by any v_i^k, while $Q_{v_i^k,v_j^k}$ captures the quality of the validation method. Since the parameters $con_{v_i^k,v_j^k}$ and $incon_{v_i^k,v_j^k}$ are not only a result of malicious intentions or faulty nodes but are also influenced by the unreliable communication links, the transmission error can contribute to the trust values of each node. With this,

$$\begin{aligned} con_{v_i^k,v_j^k} &= con_{v_j^k} + p_{tras}(con_{v_j^k}+incon_{v_j^k}), \\ incon_{v_i^k,v_j^k} &= incon_{v_j^k} - p_{tras}(con_{v_j^k}+incon_{v_j^k}), \end{aligned} \tag{13}$$

where $con_{v_j^k}$ and $incon_{v_j^k}$ receptively represent the number of consistent and inconsistent validation decisions made by the validator v_j^k. The transmission error rate [30] is obtained following

$$p_{tras} = 1 - \frac{\sum_i^n \omega(i)\times\omega(i)}{\sum_i^n \omega(i)}, \tag{14}$$

where $\omega(i)$ is the weight of the historical link-state, with $\aleph = (\omega(1),\omega(2),...,\omega(n))$ representing the historical link-state record and $\omega(i) = \frac{2i}{n(n+1)}$. The average aggregated direct trust $D^{dir}_{v_i^k,v_j^k}$ is thus obtained from $RD_{v_i^k,v_j^k}$ as

$$D^{dir}_{v_i^k,v_j^k} = \frac{\sum_n RD_{v_i^k,v_j^k}(n)}{n}. \tag{15}$$

C. Indirect trust

For the indirect trust values, any node (for instance, the primary) can obtain indirect trust values of other validators from the blockchain system or other neighbouring nodes. Similarly, the blockchain system can evaluate the reputations of some nodes through indirect trust. Since every validator forwards the trust values of their communicating pairs to the blockchain system after every observation period, the aggregated indirect trust value of each validator is always available to improve the validation decision at every observation time. To estimate the indirect trust value, suppose $D^{dir}_{v_i^k,v_l^k}$ and $D^{dir}_{v_j^k,v_l^k}$ are respectively the average aggregated direct trust values of nodes v_i^k and v_j^k about node v_l^k. Then, the collective trust values of nodes v_i^k and v_j^k about node v_l^k is given as

$$D^{v_i^k,v_j^k}_{v_l^k} = D^{dir}_{v_i^k,v_l^k} \bigoplus D^{dir}_{v_j^k,v_l^k} = (b^{v_i^k,v_j^k}_{v_l^k}, d^{v_i^k,v_j^k}_{v_l^k}, v^{v_i^k,v_j^k}_{v_l^k}), \tag{16}$$

where

$$\begin{cases} b^{v_i^k,v_j^k}_{v_l^k} = (b_{v_i^k,v_l^k}v_{v_j^k,v_l^k} + b_{v_j^k,v_l^k}v_{v_i^k,v_l^k})/q \\ d^{v_i^k,v_j^k}_{v_l^k} = (d_{v_i^k,v_l^k}v_{v_j^k,v_l^k} + d_{v_j^k,v_l^k}v_{v_i^k,v_l^k})/q \\ v^{v_i^k,v_j^k}_{v_l^k} = (v_{v_j^k,v_l^k}v_{v_i^k,v_l^k})/q \end{cases} \tag{17}$$

and

$$q = v_{v_j^k,v_l^k} + v_{v_i^k,v_l^k} - v_{v_j^k,v_l^k}v_{v_i^k,v_l^k}, \forall q \neq 0. \tag{18}$$

Similarly, given that $D^{dir}_{v_i^k,v_j^k}$ and $D^{dir}_{v_j^k,v_l^k}$ represent the direct trust values of validator v_i^k for validator v_j^k and validator v_j^k for validator v_l^k, respectively. We can obtain the indirect trust value of validator v_i^k for validator v_l^k as

$$D^{v_i^k,v_j^k}_{v_l^k} = D^{dir}_{v_i^k,v_j^k} \bigotimes D^{dir}_{v_j^k,v_l^k} = (b^{v_i^k,v_j^k}_{v_l^k}, d^{v_i^k,v_j^k}_{v_l^k}, v^{v_i^k,v_j^k}_{v_l^k}), \tag{19}$$

where

$$\begin{cases} b^{v_i^k,v_j^k}_{v_l^k} = b_{v_i^k,v_j^k}b_{v_j^k,v_l^k} \\ d^{v_i^k,v_j^k}_{v_l^k} = b_{v_i^k,v_j^k}d_{v_j^k,v_l^k} \\ v^{v_i^k,v_j^k}_{v_l^k} = d_{v_i^k,v_j^k} + v_{v_i^k,v_j^k} + b_{v_i^k,v_j^k}v_{v_j^k,v_l^k}. \end{cases} \tag{20}$$

The indirect trust value $D^{ind}_{v_i^k,v_l^k}$ is obtained following the same method as in (15). To investigate the reliability of recommendation in (19), let $D^{ave}_{v_l^k}$ represent the average value of all received recommendations for v_l^k. Then we can obtain the difference between $D^{v_i^k,v_j^k}_{v_l^k}$ and $D^{ave}_{v_l^k}$. The greater the difference, the lower the reliability of the recommendation received from any validator. Therefore, the recommendation reliability is given as

$$RelD^{v_i^k,v_j^k}_{v_l^k} = 1 - |D^{v_i^k,v_j^k}_{v_l^k} - D^{ave}_{v_l^k}|. \tag{21}$$

Blockchain continuously penalizes validators with inconsistent recommendations and may lead to removal from the system. Finally, the reputation is obtained as a function of both direct and indirect trusts, such that

$$D^{\text{trust}}_{v_i^k,v_j^k} = \omega_{dir}D^{dir}_{v_i^k,v_j^k} + \omega_{ind}D^{ind}_{v_i^k,v_j^k}, \tag{22}$$

where ω_{dir} and ω_{ind} are the weights of direct and indirect trust values respectively, given that $\omega_{dir}+\omega_{ind} = 1$. As an abuse of notation, let $D^{\text{trust}}_{v_i}$ represent the average aggregate reputation value of each validator $v_i \in V$. To maintain the security level, each validator with reputation value below $D^{\text{trust}}_{v_i} < a_{th}$ is removed from the system, such that f is defined as

$$f = \sum_{i=1}^{N} 1_{[a_{th} \leq D^{\text{trust}}_{v_i} < d_{th}]}, \tag{23}$$

where $1_{[.]}$ is an indicator function that is equal to 1 if [.] is true and 0 otherwise. The parameter d_{th} is always selected such that the intra-shard bound

$$f_k \leq \left\lceil N_{s,k} - \sum_{i=1}^{Ns,k} D^{\text{trust}}_{v_i^k}(i) \right\rceil, \tag{24}$$

where $\lceil . \rceil$ is the ceil function. To prevent the single-shard takeover, the number of shards at any PBFT view is bounded by the constraint

$$K \leq \frac{N}{3(N - \sum_{i=1}^{N} 1_{[D_{v_i}^{\text{trust}} \geq d_{th}]}) + 1}. \tag{25}$$

From (25), it is clear that the possible number of shards is limited by the reputations of all available validators. Hence, a shard is only created, following Algorithm 1, if the security constraints are satisfied. Note that the proposed framework relies on the PBFT consensus protocol and the shard formation process is subject to the constraint in (24). Hence, the number of shards K to guarantee a successful consensus for all shards while preventing appending a malicious block to the chain is captured by (25). The expression in (25) captured the worst-case scenario where all malicious nodes K, as in (23), are in the same shard. Generally, K is inversely proportional to f.

In a broad context, where the reputation value of each node $D_{v_i}^{\text{trust}}$ reflects its likelihood of being malicious or faulty, we can obtain the failure probability of any typical shard k – the probability that shard k will fail to reach an honest consensus thus will append a malicious block to the chain – as a function of the aggregated reputation values of $v_i^k \in V_k$. Since all validators operate independently, the probability of shard failure can be obtained as

$$P_f^{(k)} = 1 - \frac{1}{N_{s,k}} \sum_{i=1}^{N_{s,k}} D_{v_i^k}^{\text{trust}}. \tag{26}$$

V. Performance Analysis

In this section, we analyze some performance metrics of interest for the sBeDS framework. We evaluate scalability and consider the block generation and consensus process time.

A. Scalability

Scalability is an important metric when characterizing blockchain-enabled data-sharing framework in HDT. It measures the number of transactions that can be processed per second. This transaction throughput can be improved by either increasing the block size S^B or reducing the block interval T^I, although an increase in the S^B or a decrease in T^I can impose stricter constraints on consensus latency. Hence, the choice of appropriate method, as well as the adopted consensus algorithm, should be properly considered to obtain the trade-off between scalability and latency. The number of transactions that can be processed per second in the proposed sBeDS framework is given as

$$T_{thru}(S^B, T^I) = \frac{K\lfloor S^B/\chi \rfloor}{T^I}, \tag{27}$$

where the block interval T^I follows from (3) and represents the average time required to generate a new block. From (27), it can be observed that an increase in K can further increase the number of transactions per second, $T_{thru}(S^B, T^I)$. In a case where there is a c_{sh}^k proportion of cross-shard transactions in the kth shard, (27) is upper-bounded at

$$T_{thru}(S^B, T^I, c_{sh}^k) = \frac{K\lfloor S^B/\chi \rfloor}{T^I} - \frac{\lfloor S^B/\chi \rfloor}{2T^I} \sum_{k=1}^{K} c_{sh}^k. \tag{28}$$

In such a case, reducing c_{sh}^k increases the scalability.

B. Latency

Latency is an important metric in any blockchain-related analysis [2], [3] and can be measured as the time to finality, which is the time required to successfully append a block to the chain through any shard k. This is the same as the time until a transaction written in the blockchain is irreversible. This latency in the presented framework includes three main components: block generation time T^I, consensus time and block appending time. The time to finality can be obtained as

$$T_{TTF} = T^I + T_{con}^k + T_{app}^k, \tag{29}$$

where T_{con}^k and T_{app}^k are the consensus time and block appending time respectively for any shard k. The consensus time depends on the PBFT scheme and is obtained as

$$T_{con}^k = T_{deliv}^k + T_{val}^k, \tag{30}$$

where T_{deliv}^k and T_{val}^k are the message delivery and block validation time within any shard k respectively. To avoid unnecessary complication, we evaluated the validation time as a function of cryptographic operations computing cost similar to [19], [20], [46], where a block validation process includes signatures validation, generation and validation of MACs using ζ, η and η CPUs cycles respectively.

In the REQUEST stage, the client v_c sends a block validation request to any available primary $v_p, \forall p \neq c$, where only one MAC verification is performed. Each validation request contains one signature, which requires verification of each validator during any consensus process. During the PRE-PREPARE stage, the primary in any shard k processes a batch of M validation requests and forwards a single pre-prepare message to all replicas within shard k. In this case, the primary generates $N_{s,k} - 1$ MACs and each replica processes one MAC for verifications. Each replica then authenticates the received pre-prepare message during the PREPARE stage by generating $N_{s,k} - 1$ MACs to every other replica within shard k including the primary, while verifying $N_{s,k} - 2$ MACs. Next, each validator carries out block validation in the COMMIT stage, where all validators within shard k including the primary send and receive $N_{s,k} - 1$ commit messages, thereby generating and validating $N_{s,k} - 1$ MACs. Finally, each validator generates one MAC for each validation request to reply to the client during the REPLY stage. It becomes immediately clear that for each M validation request in any shard k, the primary processes $2M + 4(N_{s,k} - 1)$ MAC operations, while each replica processes $M + 4(N_{s,k} - 1)$ MAC operations. The validation time of a primary in any shard k is given as

$$O_{v_p}^k = \frac{M\zeta + [2M + 4(N_{s,k} + f_k - 1)]\eta}{c_{v_p}}, \tag{31}$$

where c_{v_p} is the computation capacity of the primary v_p. Similarly, the validation time of any replica is given as

$$O_{v_i}^k = \frac{M\zeta + [M + 4(N_{s,k} + f_k - 1)]\eta}{c_{v_i}}, \tag{32}$$

where c_{v_i} is the computation capacity of any validator v_i. The total validation time T^k_{val} can thus be obtained as

$$T^k_{val} = \frac{1}{M} \max_{v^k_i \in V_k} \{O^k_{v_p}, O^k_{v_i}\}. \tag{33}$$

Similarly, the message delivery time T^k_{deliv} depends on the total time to transmit a block from the client to the primary and the total message exchanging time during validation. From (9), the time to transmit a block from the client to the primary is given as

$$\varphi_{v_c, v^k_p} = \frac{MS^B}{R_{v_c, v^k_p}}. \tag{34}$$

From (34), we can obtain the message delivery time T^k_{deliv}, given that τ is the timeout, as

$$\begin{aligned} T^k_{deliv} &= \frac{1}{M}(T^k_{request} + T^k_{pre-prepare} + T^k_{prepare} + \\ &\quad T^k_{commit} + T^k_{reply}) \\ &= \frac{1}{M}\Big(\min\{\varphi_{v_c, v^k_p}, \tau\} + \min\Big\{ \max_{v^k_i \neq v^k_p, v_c} \varphi_{v^k_p, v^k_i}, \tau \Big\} + \\ &\quad \min\Big\{ \max_{v^k_i \neq v^k_j; v^k_i, v^k_j \neq v_c} \varphi_{v^k_i, v^k_j}, \tau \Big\} + \\ &\quad \min\Big\{ \max_{v^k_i \neq v^k_j; v^k_i, v^k_j \neq v_c} \varphi_{v^k_i, v^k_j}, \tau \Big\} + \min\Big\{ \max_{v^k_i \neq v_c} \varphi_{v^k_i, v_c}, \tau \Big\} \Big). \end{aligned} \tag{35}$$

From (35), the consensus time T^k_{cons} is obtained.

C. Block appending time

Since the validation process in each shard is independent of the validation process in other shards, two or more shards may complete the validation of a block at the same time slot. To ensure an efficient appending process, each shard is assigned a different priority such that the block appending process is based on the non-preemptive priority of each shard. Thus, the block appending process eliminates the possibility of forking attacks. We modeled the block appending process as a Geo/G/1 queuing system with non-preemptive priority, where any shard k has non-preemptive priority over shard $k+m, 0 < m \leq K-1$, such that the priority of blocks from shard $1 > 2 > K-1 > K$. The arrival of validated blocks from each shard, therefore, follows an independent Bernoulli process with a probability λ_k, while each validated block requires a general appending service with service probability μ_k. Under the considered stable condition and at any time slot,

$$\rho = \sum_{k=1}^{K} \frac{\lambda_k}{\mu_k} < 1. \tag{36}$$

Following the proposed priority-based block appending technique, the block from shard 1 is appended to the chain before the block from shard 2, while the block from shard 2 is appended to the chain before the block from shard 3, etc. As in real-life systems, the appending time of any block from shard 1 is not affected by the appending time of blocks from lower priority shards $k > 1$, while the appending time of any block from shard 2 is only affected by the appending time of blocks from higher priority shard 1. It follows that the appending time of blocks from any shard k is only affected by the appending time of blocks from higher priority shards $m > k$. Thus, the proposed block appending process can be captured for two special classes: higher priority class and lower priority class. The block appending time of a block from the highest priority shard $k = 1$ at any time slot can be calculated as

$$T^1_{app} = \frac{1}{2\mu_1} + \frac{\varpi_{\lambda_1}\frac{1}{\mu_1} + \lambda_1^2 \varpi_{b_1}}{2\lambda_1(1-\rho_1)} + \frac{\lambda_2\Big(\varpi_{b_1} + \frac{1}{\mu_2}\Big[\frac{1}{\mu_2} - 1\Big]\Big)}{2(1-\rho_1)}, \tag{37}$$

where ϖ_* is the variance of $*$, while $b_* = \frac{1}{\mu_*}$. Similarly, the block appending time of any block from any lower priority shard (say $k = 2$) can be obtained following

$$\begin{aligned} T^2_{app} &= \frac{1}{2\mu_2} + \frac{\varpi_{\lambda_2}\frac{1}{\mu_2}}{2\lambda_2(1-\rho)} + \frac{\lambda_2 \varpi_{b_2}}{2(1-\rho)(1-\rho_h)} \\ &\quad + \frac{\varpi_{\lambda_h}\Big(\frac{1}{\mu_h}\Big)^2 + \lambda_1 \varpi_{b_h}}{2(1-\rho)(1-\rho_h)}. \end{aligned} \tag{38}$$

The parameters λ_h and μ_h in (38) capture the joint arrival and service probabilities of blocks from higher priority shards respectively. The proof of (37) and (38) follows from the analysis of the discrete-time single server queueing system provided in [47]. At any slot, the average appending time can thus be approximated as

$$T^{ave}_{app} = \frac{1}{K} \sum_{k=1}^{K} T^k_{app}. \tag{39}$$

To ensure that the latency requirements of the blockchain-enabled data-sharing system are satisfied, the total latency should be within some consecutive block intervals ξ ($\xi > 1$), such that

$$T_{TTF} \leq \xi T^I, k = 1, ..., K. \tag{40}$$

VI. Performance optimization using DRL

In this section, we first formulate the proposed sBeDS framework as a MDP to allow optimization of transaction throughput and minimization of overall latency, while ensuring that the security constraints are satisfied as in conventional PBFT consensus protocol-based systems. Next, we introduce DRL to capture and address the dynamic nature of the proposed system.

A. State space and transition probability

By formulating the shard-based blockchain-enabled data-sharing system as a discrete MDP, we can maximize the system reward such that the MDP is defined using the tuple $(\mathcal{S}^{(t)}, \mathcal{A}^{(t)}, \mathcal{P}^{(t)}, \mathcal{R}^{(t)})$, where $\mathcal{S}^{(t)}$ is the state space, $\mathcal{A}^{(t)}$ is the action space, $\mathcal{P}^{(t)}$ is the state transition probabilities and $\mathcal{R}^{(t)}$ is the reward function. Note that the DRL framework can have two phases: (i) an offline deep neural network construction phase, where the action-value function can be approximated with corresponding states and actions, and (ii) an online dynamic deep Q learning phase, which is used for action selection, system control, and dynamic network updating. Later, we present the details of the adopted BDQ algorithm.

At any decision epoch t ($t \geq 1$), the state space $\mathcal{S}^{(t)}$ is defined as the union of data achievable rate $R = \{R_{i,j}\}$, average transaction size χ, computation capacity of validators c_v, and the reputation value $D_{i,j}^{\text{trust}}$ of validators. This can be represented as

$$\mathcal{S}^{(t)} = [R, \chi, c_v, D_{i,j}^{\text{trust}}]^{(t)}. \tag{41}$$

Note that state space in (41) is continuous, thus the probability of being in a certain state can be assumed to be zero. With this, the process transition from state $s^{(t)}$ to the next state $s^{(t+1)}$ through the action $a^{(t)} \in \mathcal{A}^{(t)}$ is given as

$$Pr(s^{(t+1)}|s^{(t)}, a^{(t)}) = \int_{\mathcal{S}^{(t+1)}} \mathbb{F}(s^{(t)}, a^{(t)}, s')ds', \tag{42}$$

where $\mathbb{F}$ is the state transition probability density function.

B. Action space

The action space $\mathcal{A}^{(t)}$ at any decision epoch t includes the offloading decision $a = \{a_n\}$, $a_n \in \{0, 1\}$, block size S^B, block interval T^I, and the number of shards K^*. This is given as

$$\mathcal{A}^{(t)} = [a, S^B, T^I, K^*]^{(t)}, \tag{43}$$

where $a_n = 1$ when a validating node participates in the validation process and $a_n = 0$ otherwise. Similarly, $S^B \in \{0.2, 0.4, ..., \dot{S^B}\}$, $T^I \in \{0.5, 1, ..., \dot{T^I}\}$, and $K^* \in \{1, 2, ..., \dot{K^*}\}$, where $\dot{S^B}$, $\dot{T^I}$ and $\dot{K^*}$ are the block size limit, maximum block interval and largest shard number satisfying the security constraint respectively.

C. Reward Function

We aim to simultaneously optimize the transaction throughput and minimize the overall latency in the shard-based blockchain-enabled data-sharing system. The objective of the system is given as

$$O = \Theta_1 T_{TTF} - (1 - \Theta_1)\Theta_2 T_{thru}, \tag{44}$$

where $\Theta_1 (0 < \Theta_1 < 1)$ is a weight factor, which is useful in combining two objective functions into a single one and Θ_2 is a mapping factor that ensures two objective functions are at the same scale. From (44), the optimization problem can be obtained as

$$\begin{aligned} \min_{\mathcal{A}^{(t)}} \mathbb{E}\left[\sum_{t=0}^{+\infty} \left(\Theta_1 T_{TTF} - (1-\Theta_1)\Theta_2 T_{thru}\right)\right] \\ \text{s.t. (C1):} \quad T_{TTF} \leq \xi T^I \\ \text{(C2):} \quad f_k \leq \frac{N_{s,k} - 1}{3} \\ \text{(C3):} \quad \sum_k N_{s,k} \leq N \\ \text{(C4):} \quad \sum_k f_k \leq f \\ \text{(C5):} \quad a_n \in \{0, 1\} \\ \text{(C6):} \quad \rho < 1. \end{aligned} \tag{45}$$

The reward function is thus defined as

$$\mathcal{R}^{(t)} = \begin{cases} -O(t), & \text{if } C1 - C6 \text{ are satisfied} \\ 0, & \text{otherwise.} \end{cases} \tag{46}$$

D. Branching dueling Q-network

Because of the dynamic and large-dimensional characteristics of the proposed sBeDS problem, it is imperative to adopt the DRL technique. The large action space introduced by the proposed scheme, however, brings great challenges to the discrete-action-based DRL optimization methods since the number of actions that need to be explicitly represented in the conventional deep deterministic policy gradient (DDPG) and deep Q-network (DQN)-based agents grows exponentially with an increasing number of validators, which makes it hard to achieve convergence. To compensate for large dimensions of action space, we apply the BDQ algorithm.

BDQ is a branching variant of the dueling double deep Q-network that incorporates the action branching architecture into the DQN to decrease the number of estimated actions [33]. The advantage of the action branching architecture can be observed when solving problems in multidimensional action spaces since it is possible to optimize each action dimension with a degree of independence. BDQ enhances scalability by ensuring the linear growth of the total number of network outputs with increasing action dimensionality. The agent in BDQ can scale gracefully to environments with increasing action dimensionality and it was shown in [33] to perform competitively when compared with the conventional DDPG and other related algorithms. BDQ allows the adoption of discrete-action algorithms in DRL for domains with high-dimensional continuous or discrete action spaces. For any action dimension $d \in \{1, 2, ..., N_d\}$, each sub-action has $|\mathcal{A}_d| = \vartheta$ discrete sub-actions. The Q-value of any branch at any state $s \in \mathcal{S}$ and sub-action $a_d \in \mathcal{A}_d$ can be expressed as a function of the common state value $V(s)$ and the corresponding sub-action advantage $A_d(s, a_d)$ following

$$Q_d(s, a_d) = V(s) + \left[A_d(s, a_d) - \frac{1}{\vartheta} \sum_{a_d \in \mathcal{A}_d} A_d(s, a'_d)\right], \tag{47}$$

such that the temporal-difference target

$$y = \mathcal{R} + \gamma \frac{1}{N_d} \sum_d \bar{Q}_d \left(s', \text{argmax}_{a'_d \in \mathcal{A}_d} Q_d(s', a'_d)\right), \tag{48}$$

where parameters $\bar{Q}_d$ and γ are the branch d of the target network $\bar{Q}$ and the learning rate respectively. From (48), the loss function can be expressed as the expected value of the mean squared temporal-difference error across the branches, given as

$$L = \mathbb{E}_{(s,a^*,r,s') \sim \mathcal{D}} \left[\frac{1}{N_d} \sum_d (y_d - Q_d(s, a_d))^2\right], \tag{49}$$

where $\mathcal{D}$ is the experience replay buffer and a^* captures the joint action tuple $(a_1^*, a_2^*, ..., a_{N_d}^*)$. To preserve the magnitudes of the errors, the unified prioritization error is expressed as

$$e_r(s, a^*, r, s') = \sum_d |y_d - Q_d(s, a_d)|. \tag{50}$$

VII. Numerical Results

In this section, we present numerical results to demonstrate the performance of the proposed sBeDS framework.

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Average transaction size, χ	200 Bytes
Computation resource of each validator, c_v	[10, 30] MHz
Computing cost for validating signatures and generating/validating MACs, ζ, η	2 MHz/1 MHz
Timeout, τ	10 s
Validation interval, ξ	20
Acceptable threshold, a_{th}	0.5
Pre-defined reputation threshold, d_{th}	0.6
Bandwidth, W	1 MHz
Offloading power of each validator, $P_{v_i^k}$	1 W
Noise signal power, σ^2	10^{-9} W
Block size limit, $\dot{S^B}$	8 MB
Maximum block interval, $\dot{T^I}$	10 s
Maximum allowable shard number, $\dot{K^*}$	8

A. Simulation Parameters

To demonstrate the performance of the proposed sBeDS framework, we carried out numerical simulations, consisting of $N = 100$ validators. The simulation results focus on the performance of the PBFT and reputation-enabled shard-based consensus process with the aim of addressing issues related to the adoption of the sBeDS framework in HDT. When evaluating the message exchanging process among validators, we adopted the Rayleigh fading assumption such that the channel gain among interfering validators is an independent and identically distributed exponential fading coefficient. Except otherwise mentioned, the parameters settings selected similar to [8], [20], [48] are summarized in Table III.

For comparison, we modified the proposed sBeDS scheme to produce three existing schemes: single shard scheme, $K = 1$; fixed block interval scheme, $T^I = 7$ s; and fixed block size scheme, $S^B = 5$ MB. For simplicity, we refer to the sBeDS scheme with all parameters optimized as the proposed scheme and set $\Theta_1 = 0.9$ and $\Theta_2 = 0.001$ similar to [28], [31], [32].

B. Simulation Results

In our simulations, we used a computer system with 10 CPU cores. The CPU is Intel(R) Core(TM) i9-10900X with 3.70GHz. We used PyTorch 0.4.1 and Python 3.6.6 as the software environment. The convergence performance of the proposed scheme is presented in Fig. 2. The total reward is shown to increase with the learning process until the optimal blockchain parameters are found. The proposed multi-shard scheme achieves higher throughput and lower latency thus a higher reward is observed compared to the other schemes. Interestingly, the convergence speed is relatively low when compared to the other schemes since the adopted multi-shard approach with variable block size and interval imposes more learning tasks on the agent at the initial learning stage. Notwithstanding that, the proposed multi-shard scheme can still achieve a reasonable convergence speed, while providing a higher total reward. It is worth noting that, even at $\xi = 20$, the single shard scheme continues to provide a total reward closer to zero since such an approach suffers from low throughput and high overall latency.

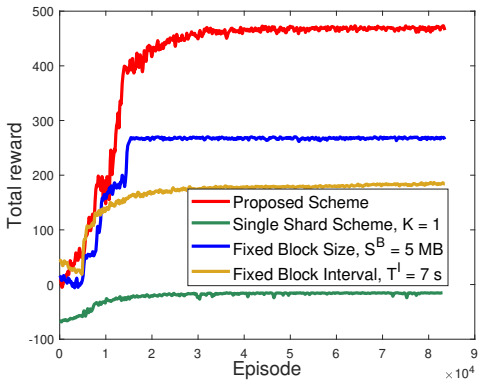

Fig. 2. Convergence performance with rewards.

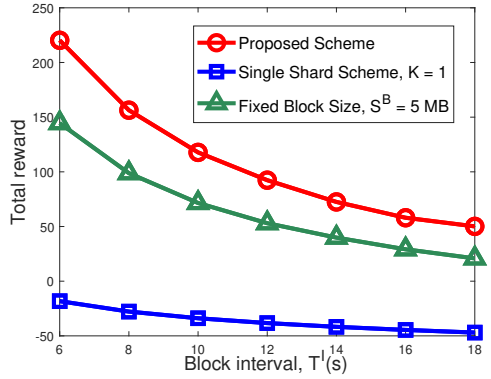

Fig. 3. Effects of block interval on the performance.

Next, we investigate the effects of the block interval on the performance of the proposed approach as presented in Fig. 3. The total reward is observed to decrease as the block interval increases since an increase in the block interval increases the overall latency while reducing the throughput. Under the fixed block interval scheme, an increase in the block interval does not affect the performance, thus a multi-shard scheme with a fixed block interval continues to produce a constant total reward. For other schemes with variable block intervals, the proposed scheme achieves a better total reward when compared with the single shard and fixed block size schemes. This further justifies that a multi-shard approach with variable block size and interval can achieve better performance. It is worth mentioning that all the considered schemes have been implemented by taking into consideration the required security constraint as in the conventional PBFT consensus protocol. Thus, a multi-shard approach not only outperforms other schemes but also achieves the same level of security.

In Fig. 4, we evaluate the effects of the average transaction size on the overall performance of the proposed scheme. As the transaction size increases, the overall performance of the proposed scheme reduces since an increase in the transaction size increases validation time, which further increases the overall latency while reducing the average throughput. Interestingly, the proposed scheme can provide better performance compared to other schemes because such a scheme can optimally adjust

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3382829

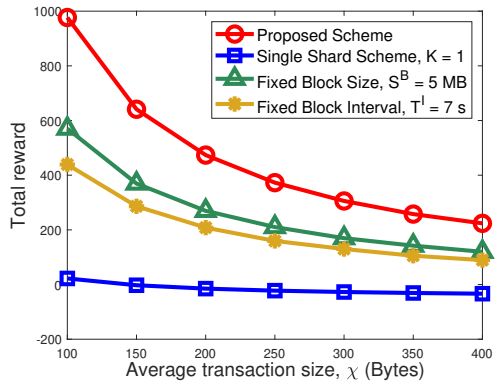

Fig. 4. Effects of transaction size on the performance.

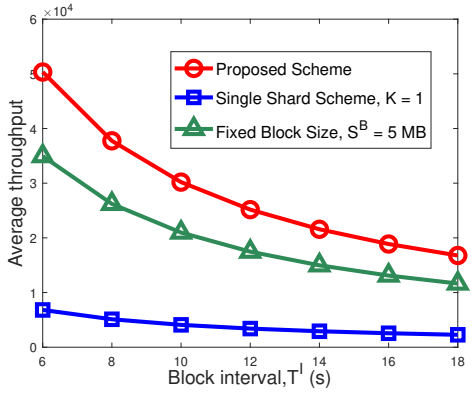

Fig. 6. Average throughput vs block interval.

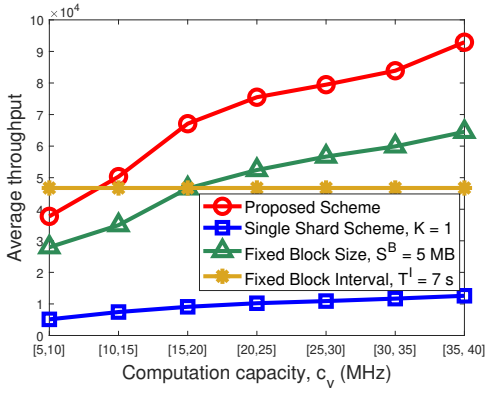

Fig. 5. Average throughput performance.

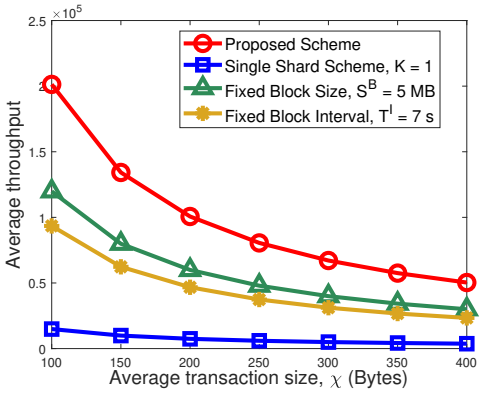

Fig. 7. Effects of transaction size on average throughput.

other blockchain parameters to compensate for the increase in the average transaction size.

To investigate the effects of the computation capacity of validators on average throughput, Fig. 5 shows that the average throughput increases as the computation capacity limit of validators increases. This is expected since an increase in c_v will improve the validation process, thereby reducing the overall latency. Similarly, the proposed scheme achieves a better average throughput compared to other schemes. For the scheme with a fixed block generation interval, the average throughput remains constant as c_v increases since an improved validation experience (i.e., reduced validation time) in such a scheme means validators will remain idle for a longer period owing to the fixed block interval. The fixed block interval scheme is expected to produce an improved performance when c_v is low as can be seen in Fig. 5.

Furthermore, Fig. 6 represents the effects of block interval on the average throughput. This is similar to Fig. 3, where the impacts of block interval on the overall performance of the proposed scheme were investigated. As expected, the average throughput reduces as the block interval increases, while the proposed scheme continues to produce better throughput. Similarly, Fig. 7 demonstrates the relationship between the average throughput and the average transaction size. The average throughput is observed to decrease as the transaction size increases since an increased transaction size further increases the validation time as well as message exchanging time among validators, which directly affects the overall latency. When compared with the single-shard scheme, the proposed multi-shard schemes achieve better performance justifying the importance of the multiple validation process towards improving the validation process of blockchain-enabled data-sharing systems.

Finally, the proposed PBFT and reputation-enabled shard-based scheme is capable of improving the scalability and throughput of blockchain-based data-sharing systems, and reducing overall latency, while ensuring sufficient decentralization and security features. As can be observed from the proposed framework, the approach is decentralized while the proposed integrated PBFT and trust-based proof of reputation scheme ensures that the security level is not compromised. Estimating the value of f based on the reputation values means only validators with acceptable historical reputations are always selected during the shard formation process thus, eliminating the probability of appending malicious blocks to the chain as in the conventional PBFT scheme, while providing an improved validation experience.

VIII. Conclusion

In this paper, we introduced a sBeDS framework designed to improve the data-sharing experience within zero-trust HDT

systems. To ensure the proposed solution maintains high scalability without compromising its decentralization and security features, we integrated trust-based proof-of-reputation and PBFT consensus techniques, as well as a priority-based queueing approach aimed at refining the block appending process. The analysis of performance metrics emphasized the significance of parallel validation in the considered blockchain-enabled data-sharing system. Furthermore, we addressed the challenges of joint transaction offloading and computation resource allocation by formulating the resulting problem as an MDP, enabling optimized throughput with simultaneous reductions in communication and computation latency. By employing the BDQ approach tailored for expansive action space dimensions, our findings highlight the effectiveness of the proposed solution. These findings contribute significantly to overall performance improvements in blockchain-enabled data-sharing, thereby promising enhanced user experiences in HDT systems and related large-scale applications.

In the future, we aim to enhance our proposed sBeDS solution by addressing potential sources of subjectivity and bias, as well as strengthening the system against vulnerabilities such as Sybil attacks. We will attempt to refine the incorporated reputation-enabled scheme to ensure a more objective and unbiased evaluation of user contributions, mitigating the impact of subjective influences on reputation scores. Simultaneously, we plan to implement advanced mechanisms to detect and counteract Sybil attacks, bolstering the system's resilience against such manipulative tactics. This, we believe, will further improve the integrity and security of the sBeDS framework, fostering a more robust and trustworthy environment for all users in HDT and many other applications.

REFERENCES

[1] J. Chen, C. Yi, S. D. Okegbile, J. Cai, and X. S. Shen, " Networking Architecture and Key Supporting Technologies for Human Digital Twin in Personalized Healthcare: A Comprehensive Survey, IEEE Communications Surveys & Tutorials, Sept. 2023, doi: 10.1109/COMST.2023.3308717.

[2] S. D. Okegbile, and J. Cai, "Edge-assisted human-to-virtual twin connectivity scheme for human digital twin frameworks," in IEEE VTC Conference, Helsinki, Jun. 2022, pp. 1–6.

[3] S. D. Okegbile, J. Cai, C. Yi, and D. Niyato, "Human Digital Twin for Personalized Healthcare: Vision, Architecture and Future Directions," IEEE Network, vol. 37, no. 2, pp. 262 – 269, Mar. 2023.

[4] Z. Lv, C. Cheng, and H. Lv, "Blockchain Based Decentralized Learning for Security in Digital Twins, IEEE Internet of Things Journal, vol. 10, no. 24, pp. 21479 – 21488, Dec. 2023.

[5] S. Qi, X. Yang, J. Yu, and J. Qi, "Blockchain-aware Rollbackable Data Access Control for IoT-enabled Digital Twin," IEEE Journal on Selected Areas in Communications, vol. 41, no. 11, pp. 3517 – 3532, Nov. 2023.

[6] S. D. Okegbile, J. Cai, H. Zheng, J. Chen, and C. Yi, "Differentially Private Federated Multi-Task Learning Framework for Enhancing Human-to-Virtual Connectivity in Human Digital Twin," IEEE Journal on Selected Areas in Communications, vol. 41, no. 11, pp. 3533 – 3547, Nov. 2023.

[7] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K. Y. Lam, and L. H. Koh, "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey," IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4157–4185, Oct. 2020.

[8] J. Yun, Y. Goh, and J. M. Chung, "DQN-Based Optimization Framework for Secure Sharded Blockchain Systems," IEEE Internet of Things Journal, vol. 8, no. 2, pp. 708–722, Jul. 2020.

[9] S. Lee, M. Kim, J. Lee, R. H. Hsu, and T. Q. Quek, "Is Blockchain Suitable for Data Freshness? An Age-of-Information Perspective," IEEE Network, vol. 35, no. 2, pp. 96–103, Feb. 2021.

[10] O. Vashchuk and R. Shuwar, "Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake," Electronics and Information Technologies, vol. 9, no. 9, pp. 106–112, Jan. 2018.

[11] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," Operation System Design Implementation, vol. 99, no. 1999, pp. 173–186, Feb. 1999.

[12] Y. Du, Z. Wang, J. Li, L. Shi, D. Jayakody, Chen, Q. Chen, W. Chen, and Z. Han, "Blockchain-Aided Edge Computing Market: Smart Contract and Consensus Mechanisms," IEEE Transactions on Mobile Computing, vol. 22, no. 6, pp. 3193 – 3208, Jun. 2023.

[13] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance Evaluation of blockchain systems: A systematic survey," IEEE Access, 8, pp. 126927–126950, Jun. 2020.

[14] M. Alaslani, F. Nawab, and B. Shihada, "Blockchain in IoT systems: End-to-end delay evaluation," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8332–8344, Oct. 2019.

[15] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in Proceedings of ACM SIGSAC Conference on Computer and Communications Security, New York, Oct. 2016, pp. 17–30.

[16] J. Ye, X. Kang, Y. C. Liang, and S. Sun, "A Trust-Centric Privacy-Preserving Blockchain for Dynamic Spectrum Management in IoT Networks," IEEE Internet of Things Journal, vol., no. 15, pp. 13263–13278, Aug. 2022.

[17] Y. Liu, K. Hao, W. Ren, R. Xiong, T. Zhu, K. Choo, and G. Min, "A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust internet-of-things," IEEE Transactions on Computers, vol. 72, no. 2, pp. 501–512, Mar. 2022.

[18] Y. Liu, X. Xing, Z. Tong, X. Lin, J. Chen, Z. Guan, Q. Wu, and W. Susilo, "Secure and Scalable Cross-Domain Data Sharing in Zero-Trust Cloud-Edge-End Environment Based on Sharding Blockchain," IEEE Transactions on Dependable and Secure Computing, Sept. 2023, doi: 10.1109/TDSC.2023.3313799.

[19] M. Liu, Y. Teng, F. R. Yu, V. C. Leung, and M. Song, "Deep reinforcement learning based performance optimization in blockchain-enabled internet of vehicle," in IEEE International Conference on Communications, Shanghai, May 2019, pp. 1–6.

[20] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach," IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3559–3570, Feb. 2019.

[21] M. Kim, S. Lee, C. Park, J. Lee, and W. Saad, "Ensuring Data Freshness for Blockchain-enabled Monitoring Networks," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9775–9788, Jun. 2022.

[22] S. D. Okegbile, J. Cai, A. S. Alfa, "Performance analysis of blockchain-enabled data sharing scheme in cloud-edge computing-based IoT networks," IEEE Internet of Things Journal, vol. 9, no. 21, pp. 21520 – 21536, Nov. 2022.

[23] J. H. Chen, M. R. Chen, G. Q. Zeng, and J. S. Weng, "BDFL: a byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle," IEEE Transactions on Vehicular Technology, vol. 70, no. 9, pp. 8639–8652, Aug. 2021.

[24] O. Alfandi, S. Otoum, and Y. Jararweh, "Blockchain solution for IOT-based critical infrastructures: Byzantine fault tolerance," in IEEE Network Operations and Management Symposium, Budapest, Jun. 2020, pp. 1–4.

[25] S. Gao, T. Yu, J. Zhu, and W. Cai, "T-PBFT: An EigenTrust-based practical Byzantine fault tolerance consensus algorithm," China Communications, vol. 16, no. 12, pp. 111–123, Dec. 2019.

[26] S. Kim, S. Lee, C. Jeong, and S. Cho, "Byzantine Fault Tolerance Based Multi-Block Consensus Algorithm for Throughput Scalability," in IEEE International Conference on Electronics, Information, and Communication, Barcelona, Jan. 2020, pp. 1–3.

[27] A. Loveless, R. Dreslinski, B. Kasikci, and L. T. Phan, "IGOR: Accelerating byzantine fault tolerance for real-time systems with eager execution," in IEEE Real-Time and Embedded Technology and Applications Symposium, Nashville, Jul. 2021, pp. 360–373.

[28] J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6214–6228, Dec. 2019.

[29] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of vehicles: Optimizing consensus management using reputation and contract theory," IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2906–2920, Jan. 2019.

[30] G. Han, J. Jiang, L. Shu, and M. Guizani, "An attack-resistant trust model based on multidimensional trust metrics in underwater acoustic

sensor network," IEEE Transactions on Mobile Computing, vol. 14, no. 12, pp. 2447–2459, Feb. 2015.
[31] X. Jiang, F. R. Yu, T. Song, and V. C. Leung, "Intelligent resource allocation for video analytics in blockchain-enabled internet of autonomous vehicles with edge computing," IEEE Internet of Things Journal, vol. 9, no. 16, pp. 14260–14272, Aug. 2022.
[32] L. Liu, J. Feng, Q. Pei, C. Chen, Y. Ming, B. Shang, and M. Dong, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor-critic learning approach," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2342–2353, Dec. 2020.
[33] A. Tavakoli, E. Pardo, and P. Kormushev, "Action branching architectures for deep reinforcement learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, April. 2018.
[34] R. Zhang, F. R. Yu, J. Liu, T. Huang, and Y. Liu, Y. "Deep reinforcement learning (DRL)-based device-to-device (D2D) caching with blockchain and mobile edge computing," IEEE Transactions on Wireless Communications, vol. 19, no. 10, pp. 6469–6485, Jun. 2020.
[35] P. Zhang, W. Guo, Z. Liu, M. Zhou, B. Huang, and K. Sedraoui, "Optimized Blockchain Sharding Model Based on Node Trust and Allocation," IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 2804 – 2816, Sept. 2023.
[36] X. Huang, Y. Wang, Q. Chen, and J. Zhang, "Security Analyze with Malicious Nodes in Sharding Blockchain based Fog Computing Networks," in IEEE Vehicular Technology Conference, Norman, Sept. 2021, pp. 1–5.
[37] Z. Yang, R. Yang, F. R. Yu, M. Li, Y. Zhang, and Y. Teng, "Sharded Blockchain for Collaborative Computing in the Internet of Things: Combined of Dynamic Clustering and Deep Reinforcement Learning Approach," IEEE Internet of Things Journal, vol. 9, no. 17, pp. 16494–16509, Sept. 2022.
[38] N. Gao, R. Huo, S. Wang, T. Huang, and Y. Liu, "Sharding-Hashgraph: A High Performance Blockchain-Based Framework for Industrial Internet of Things with Hashgraph Mechanism," IEEE Internet of Things Journal, vol. 9, no. 18, pp. 17070–17079, Sept. 2022.
[39] J. Xi, G. Xu, S. Zou, Y. Lu, G. Li, J. Xu, and R. Wang, "A blockchain dynamic sharding scheme based on hidden Markov model in collaborative IoT," IEEE Internet of Things Journal, volume. 10, no. 16, pp. 14896 – 14907, August. 2023.
[40] Z. Cui, Z. Xue, Y. Ma, X. Cai, and J. Chen, "A many-objective optimized sharding scheme for blockchain performance improvement in end-edge enabled internet of things," IEEE Internet of Things Journal, vol. 10, no. 24, pp 21443 – 21456, Dec. 2023.
[41] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "A multi-class channel access scheme for cognitive edge computing-based internet of things networks," IEEE Transactions on Vehicular Technology, vol. 71, no. 9, pp. 9912-9924, May 2022.
[42] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "A multi-user tasks offloading scheme for integrated edge-fog-cloud computing environments," IEEE Transactions on Vehicular Technology, vol. 71, no. 7, April 2022.
[43] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "Interference characterization in underlay cognitive networks with intra-network and inter-network dependence," IEEE Transactions on Mobile Computing, vol. 20, no. 10, pp. 2977–2991, Oct. 2021.
[44] V. K. Garg, Y. Narahari, and M. N. Murty, "Novel biobjective clustering (BiGC) based on cooperative game theory," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, pp. 1070–1082, Apr. 2012.
[45] J. Wang, and H. Wang, "Monoxide: Scale out blockchains with synchronous consensus zones," in Proc. of Networked Systems Design and Implementation, 2019, pp. 95–112.
[46] A. Clement, E. Wong, L. Alvisi, and M. Dahlin, "Making Byzantine fault tolerant systems tolerate Byzantine faults," in Proceeding of Networked Systems Design and Implementation, Boston, Apr. 2009, pp. 153–168.
[47] J. Walraevens, D. Fiems, and H. Bruneel, "Performance analysis of priority queueing systems in discrete time," in Network Performance Engineering. Berlin, Germany: Springer, 2011, pp. 203–232.
[48] S. D. Okegbile, J. Cai, and A. S. Alfa, "Practical Byzantine Fault Tolerance-Enhanced Blockchain-Enabled Data Sharing System: Latency and Age of Data Package Analysis," IEEE Transactions on Mobile Computing, vol. 23, no. 1, pp. 737 – 753, Nov. 2022.

Samuel D. Okegbile received the Ph.D. degree in computer engineering from the University of Pretoria, Pretoria, South Africa, in 2021. He is currently a Post-Doctoral Fellow with the Network Intelligence and Innovation Laboratory, Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada. His research interests include pervasive and mobile computing which includes various interesting topics in the human digital twin, the Internet of Things, data sharing, artificial intelligence, wireless communication networks, and blockchain. He has received several awards, including the Horizon Postdoctoral Scholarship, the SENTECH Scholarship, and the University of Pretoria Doctoral Scholarship. He served as the Publication Chair for the 2023 Biennial Symposium on Communications. He is also a regular reviewer of some IEEE journals and conferences.

Jun Cai (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 2004. From 2004 to 2006, he was a Post-Doctoral Fellow with the Natural Sciences and Engineering Research Council of Canada (NSERC), McMaster University, Canada. From 2006 to 2018, he was with the Department of Electrical and Computer Engineering, University of Manitoba, Canada, where he was a Full Professor and the NSERC Industrial Research Chair. In 2019, he joined the Department of Electrical and Computer Engineering, Concordia University, Canada, as a Full Professor, and the PERFORM Centre Research Chair. His current research interests include edge/fog computing, eHealth, radio resource management in wireless communications networks, and performance analysis. He received the Best Paper Award from Chinacom in 2013, the Rh Award for outstanding contributions to research in applied sciences from the University of Manitoba in 2012, and the Outstanding Service Award from the IEEE GLOBECOM 2010. He served as the Registration Chair for QShine 2005, the Track/Symposium Technical Program Committee (TPC) Co-Chair for the IWCMC 2008, the IEEE GLOBECOM 2010, the IEEE VTC 2012, the IEEE CCECE 2017, and the IEEE VTC 2019, the Publicity Co-Chair for the IWCMC 2010, 2011, 2013, 2014, 2015, 2017, and 2020, the TPC Co-Chair for the IEEE GreenCom 2018, and the General Chair for the 2023 Biennial Symposium on Communications. He also served on the editorial board for the IEEE INTERNET OF THINGS JOURNAL, IET Communications, and Wireless Communications and Mobile Computing.

Jiayuan Chen received the M.S. degree from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he is currently pursuing the Ph.D. degree in computer science and technology. His research interests include reinforcement learning, mechanism design, and distributionally robust optimization with applications in resource management and decision-making for edge computing, digital twins, and artificial intelligence-generated content.

Changyan Yi received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Manitoba, MB, Canada, in 2018. He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA) and also affiliated with the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China. From September 2018 to August 2019, he worked as a Research Associate with the University of Manitoba. His research interests include game theory, queueing theory, and machine learning and their applications in various wireless networks, including edge/fog computing, IoT, and 5G and beyond. He was awarded the Changkong Scholar of NUAA in 2018 and the Chinese Government Award for Outstanding Students Abroad in 2017.