# A Novel Reputation-based Sharding Blockchain System in Edge Sensor Networks

Jiarui Zhang, Yuanyuan Yang
Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA
{jiarui.zhang.2, yuanyuan.yang}@stonybook.edu

*Abstract*—**Edge computing enables widely distributed nodes with varying sensing, storage, and computing resources to collaborate effectively. For instance, sensor nodes gather data, store it in storage nodes, and allow computing nodes to access it as needed. However, the heterogeneity of edge computing poses a significant challenge in designing a mechanism to match the diverse and widely deployed devices. Additionally, the vast number of edge devices and their limited computing power make it impractical for a single device to maintain information for the entire network. This paper introduces a blockchain mechanism for large-scale data collection and management in edge networks. We propose a consensus mechanism for nodes in edge networks, enabling interactive clients to evaluate the quality of services provided by sensors. To alleviate the burden on edge devices in maintaining the blockchain, we employ sharding to divide the blockchain into multiple committees, thereby reducing the maintenance load. We also outline the design of the corresponding blockchain structure to adapt our proposed mechanism. Extensive simulation results demonstrate that our mechanism can save on-chain storage, improve data quality, and verify reputation.**

## I. Introduction

Edge sensor networks represent an evolution in data collection and processing, pushing the boundaries of traditional sensor networks by leveraging edge computing [1]. Given that a single device cannot handle increasingly large-scale tasks, emerging edge computing has facilitated the cooperation of edge devices with varying capabilities globally. With the exponential growth of data generated by the proliferation of Internet-of-Things (IoT) devices, sensors, and other data-generating sources, the ability to process and analyze this data in real-time is critical [2], [3]. Edge devices are increasingly storing and accessing data in various scenarios. Medical sensors, for instance, are employed to monitor the physical conditions of people [4]. Mobile video streaming relies on base stations to store and process video streaming [5]. Vehicles use cellular networks to access maps and real-time traffic information for route planning [6]. These scenarios necessitate servers that can store large amounts of data generated within the network and provide quick access to this data when needed by end devices [7].

Traditional centralized solutions often require a foundational level of trust, such as reliance on third-party central servers, certificates, or transitive trust assumptions. These centralized solutions have significant and inevitable limitations, including efficiency issues caused by disconnections and privacy concerns stemming from the need for trust. To mitigate these problems, decentralized solutions are introduced into the edge computing environment. A prominent example is blockchain [8], which serves as a secure and effective decentralized consensus technology. Blockchain functions as a distributed ledger and is the most popular decentralized system in peer-to-peer (P2P) networks. In the storage process, an edge device acquires new data, stores it on other edge devices or servers within the edge network, and records the storage location on the blockchain. During the access process, the edge device queries the blockchain to find the data's storage location and directly selects the appropriate location for the request. There are already numerous decentralized storage systems [9], such as Storj [10], Sia [11], and Filecoin [12], which utilize popular decentralized storage mechanisms. Compared to centralized solutions, blockchain solutions ensure that all content can be accessed even if some nodes are offline, and data storage is traceable and immune to malicious tampering by adversaries.

The current solutions in edge sensor network environments still have two main drawbacks. Firstly, sensors are heterogeneous, leading to variations in the quality of services or data they provide. Unstable network connections can also affect the reliability of participants. Hence, it is crucial to evaluate sensors, allowing users to refer to historical data and assessments to request better services and resources. A reputation mechanism is a suitable solution for sensor evaluation [13]. Starting from e-commerce systems, centralized reputation systems have seen significant development [14]. However, decentralized reputation systems [15], [16] eliminate biases that may arise from centralization, becoming a more popular paradigm. Artificial intelligence can also learn about reputation [17], but the large models required for learning are too costly for sensors. Although reputation has been studied widely, no system is currently suitable for maintaining the reputation of a large number of sensors in a heterogeneous environment. After establishing a reputation system, participants in the network need to reach a consensus on reputation. The classic consensus mechanism used by traditional decentralized blockchain consensus is Proof-of-Work (PoW), which consumes significant computing resources. Given that most edge devices have limited computing power, they cannot afford the computational expense required to generate blocks. Some work [18], [19] suggests that edge devices could purchase computing power from servers to mine, but this may not be desirable due to additional costs. Other mechanisms, such as Proof-of-Stake (PoS) [20], [21] and Proof-of-Storage Time (PoSt) [22], [23], allocate consensus subjects based on the resources they possess. However, these mechanisms have additional requirements for network elements and may lead to inevitable centralization. Therefore, finding a suitable consensus mechanism is crucial to address this issue effectively. When addressing the sensor reputation problem in edge networks, several challenges arise: 1) How to establish an appropriate reputation mechanism for sensors to manage service quality; 2) How to design a consensus mechanism suitable for limited resources; 3) How to create an efficient blockchain structure based on the specific characteristics of the problem. Customizing a reputation mechanism for environments with heterogeneous sensors and designing an appropriate consensus mechanism are essential steps. Moreover, developing a blockchain structure that efficiently supports the reputation mechanism and consensus process is critical for the successful implementation of a reliable and effective edge network solution.

In this paper, we first design a reputation mechanism for an edge sensor network. Each client in the network shares the reputation of other sensors, and the reputation of each client is determined by the overall quality of the sensors it manages. Leveraging this reputation mechanism, we develop a novel sharding blockchain to maintain information about sensors and the data they collect. Blockchain sharding enhances the scalability of the blockchain, allowing it to support more clients and sensors. To integrate our reputation mechanism and sharding blockchain, we propose a reputation-based sharding blockchain structure. This structure assists nodes in managing sensor reputation, requesting data collected by sensors, and reaching a consensus on the blockchain content. Extensive simulations demonstrate that our proposed system can support a large number of sensors in the network and ensure network consensus with minimal additional consumption.

Our main contributions are summarized as follows.

- We design a reputation mechanism for clients in edge networks to evaluate sensors and other clients. Based on client-to-sensor evaluations, the mechanism derives aggregated sensor reputations and aggregated client reputations.
- We apply the sharding technique to reduce the data spread across the entire network. Clients are divided into committees to effectively maintain the reputations of sensors and clients.
- We build a blockchain to maintain the reputation-based mechanism. We specify the structure of the blocks, and the blockchain consensus mechanism ensures low cost and decentralization.
- We conduct simulations in edge networks to assess the performance of our proposed system. The simulation results demonstrate that the performance of our sharding blockchain system meets expectations across various metrics.

The rest of this paper is organized as follows. Section II briefly reviews the related work. In Section III, we formulate the problem. In Section IV, we introduce the reputation mechanism. We demonstrate the sharding mechanism in Section V. In Section VI, we propose the blockchain structure. Simulation results are presented in Section VII. The last section concludes this paper.

## II. Related Work

Bitcoin was invented in 2008 by Satoshi Nakamoto [8] and has been one of the most popular P2P applications. The related technology blockchain is composed of a chain of blocks, and the blockchain records transactions generated by P2P network participants as a distributed consensus ledger. Briefly speaking, blockchain was first designed for anonymous cryptocurrency transactions in untrusted environments, and nodes can join a permissionless network freely.

The advantages of Proof-of-Reputation (PoR), including low energy consumption and safety performance, make it an emerging blockchain consensus mechanism. PoR blockchain applications covering a variety of scenarios are rising. Gai et al. [24] presented PoR, the reputation serves as the incentive for both good behavior and block publication. Based on PoR, Yu et al. invented RepuCoin [25] with better attack resistance and higher throughput compared to PoW blockchains. Oliveira et al. [26] developed PoR and proposed an advanced consensus for private blockchain systems. Wang et al. [27] implemented a reputation incentive scheme for blockchain consensus of Industrial Internet of Things (IIoT).

Blockchain techniques provide secure and decentralized applications in edge computing environments. To empower resource trading in mobile edge computing networks, Qiao et al. [28] applied blockchain to manage resource trading and task assignment. It combines a third-party trust center server and blockchain ledger to manage activities reliably. Huang et al. [29] proposed a consensus resource allocation system in pervasive edge computing environments. This work fairly and efficiently allocates storage resources and applies PoS to save energy consumption. As practical applications, edge computing and blockchain have become common tools for Internet-of-Vehicles (IoV) [30], [31].

Reputation management in different edge computing scenarios draws attention from researchers in recent years. Some work that applies reputation mechanisms to blockchain is studied based on various application scenarios. Huang et al. [32] discussed reputation management in vehicular edge computing and networks. Liu et al. [33] optimized resource allocation in blockchain-based video streaming systems with mobile edge computing. With the help of reinforcement learning, Xiao et al. [34] utilized a blockchain-based trust mechanism to resist attacks in edge networks. In edge computing and networks, the trust environment, conditional restrictions, and environmental requirements are highly customized, thus we can see a variety of different reputation mechanisms and blockchain designs.

## III. Problem Formulation

### A. Problem Statement

Given the heterogeneous computing resources of sensors, we assume that sensors primarily perform simple functions: collecting data for their designed purposes and transmitting data to participants who need it. To manage sensors and collect their data, we use the term "client" to represent users with these functions. Sensors with sufficient computing resources to perform complex functions can be considered both individual sensors and individual clients. In the context of edge computing with limited capacity, each sensor must be bonded to a specific client. A client can bond with multiple sensors and is responsible for collecting and managing data from these sensors.

To characterize the sensor collection system model, we describe two operations that nodes in the network may perform. First, a node collects data from the sensors and stores it in cloud storage. Second, a node requests a specific piece of data from the cloud storage. Given varying performances and data quality of sensors, clients may have different evaluations of different sensors based on the data collected.

To help clients quantify the reliability of sensors, we design a reputation mechanism that enables clients to evaluate the reputations of sensors. We structure this mechanism so that clients in edge networks can assess data and historical reputation records. Based on sensor performance, we compute the aggregated reputations of the responsible clients for consensus. Given the potentially large number of sensors and clients, we apply a sharding mechanism to alleviate the pressure on computational and storage resources for clients. Our focus centers on the following three issues.

1) We design a reputation mechanism for sensors in the network, enabling interacting clients to provide evaluations for reference.
2) We propose a sharding mechanism for clients in the network to reduce on-chain costs and resolve conflicts in the reputation mechanism.
3) We build a structure to maintain and support consensus on storage allocation and the reputation mechanism.

### B. Network Assumptions

The network is composed of clients $\mathcal{C} = \{c_i\}_{i=1}^{C}$ and sensors $\mathcal{S} = \{s_j\}_{j=1}^{F}$. Clients are responsible for collecting data from sensors, storing data in cloud storage, and requesting data from cloud storage. Each facility in the network has varying storage resources. We assume that all clients possess the basic computing power required to generate blockchain blocks. Additionally, clients are expected to pay for cloud storage services, both for storing and requesting data. This payment mechanism helps deter clients from making malicious data requests and incentivizes cloud storage providers to offer their services. The specifics of the payment method are beyond the scope of this paper.

In this paper, we assume that cloud storage providers have sufficient capacity to store the collected data and act honestly. Compared to sensors and clients in edge computing environments, participants in decentralized storage systems generally possess significantly greater storage capabilities [35]. Even within decentralized systems, large corporations might still influence the mining process [36]. However, given the costs associated with maintaining such storage systems, there are no substantial financial incentives for manipulating public data collected for edge services. Consequently, we assume that storage providers operate with integrity.

To represent the relationship between clients and sensors, we use the indicator variable $b_{ij}$, where $b_{ij} = 1$ if client $c_i$ is bonded with sensor $s_j$, and $b_{ij} = 0$ otherwise. Each sensor can be bonded to only one client, so for every sensor $s_j$, we have the constraint $\sum_i b_{ij} = 1$. For simplicity, once bonded, a sensor cannot change its client. If a change is necessary, the sensor would need to cease its service and create a new identity to join the network with a different client.

## IV. Reputation Mechanism

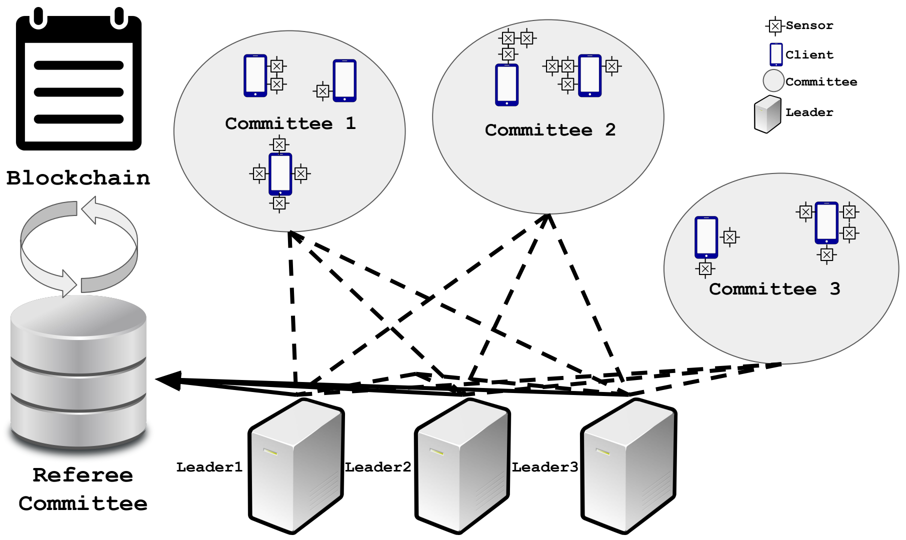
In this section, we introduce our reputation mechanism, designed for clients to evaluate sensors. The reputation of a sensor reflects how other nodes assess its behavior based on data quality and performance. Due to issues such as physical damage or transmission loss, sensors may provide varying levels of data quality. Our reputation mechanism allows clients to assess sensors based on the quality of data they deliver. Generally, sensors with higher reputations adhere more closely to expected behavior compared to those with lower reputations. However, evaluations can vary among clients. The reputation mechanism aggregates these evaluations to compute a comprehensive reputation score for each sensor. Additionally, the reputation of a client is influenced by the quality of the sensors it manages.

### A. Sensor Reputation

*1) Personal Sensor reputation:* A client $c_i$ evaluates a personal sensor reputation $p_{ij}$ for sensor $s_j$. This personal sensor reputation is derived from the perspective of $c_i$ based on the data from $s_j$. Different clients may observe different historical behaviors due to different perspectives, and they may have different personal sensor reputation evaluations of the same sensor. Each client proposes its personal sensor reputation and cannot modify the reputation assessed by other clients. In other words, client $c_i$ maintains its personal sensor reputation $p_{ij}$ for facility $j$, and only $i$ has the authority to update $p_{ij}$.

*2) Formulated Evaluations:* Every time a client $c_i$ updates a personal sensor reputation $p_{ij}$ for sensor $s_j$ is counted as a one-time evaluation. However, for further aggregation, simply aggregating all up-to-date personal sensor reputations from all clients is not proper. There are two reasons. First, the evaluation time needs to be considered. Simply regarding all evaluated personal sensor reputation up-to-date is not comprehensive, because the performances of the sensor may fluctuate, and the up-to-date personal sensor reputation from one client cannot reveal the performance over a period of time. Second, the contribution from clients cannot be equaled, as clients that request more data have a larger share in the aggregation.

Formally, an evaluation $e_k \in \mathcal{E}$ can be represented by a tuple $(c_i, s_j, p_{ij}, t_{ij})$, while $c_i$ is the client, $s_j$ is the sensor, $p_{ij}$ is the personal sensor reputation, and $t_{ij}$ is the latest evaluation



(a)

Fig. 1. A simple graph illustrates the structure of our environment. Each sensor is bonded to a client. Clients are organized into committees, with one client from each committee selected as the committee leader. A separate referee committee supervises the performance of these leaders and records essential information on the blockchain.

time. Specifically, the latest evaluation time is indicated by the block height.

*3) Standarized Sensor Reputation:* Since the evaluation criteria and dimensions of each client are different, we standardize the personal sensor reputation by EigenTrust [37] to scale sensor reputations as shown below,

$$p'_{ij} = \frac{\max(p_{ij}, 0)}{\sum_j \max(p_{ij}, 0)}. \tag{1}$$

Each $p_{ij}$ is the up-to-date personal sensor reputation given by client $c_i$. As a result, sensor reputation $p_{ij}$ is standardized to $p'_{ij}$ for each $c_i \in \mathcal{C}, s_j \in \mathcal{S}$. Since this is a generalized process, in our following discussions, we present all standardized $p'_{ij}$ as $p_{ij}$ with no ambiguity. In other words, we assume that all $p_{ij}$ is standardized.

*4) Aggregated Sensor Reputation:* We calculate the aggregated sensor reputation of sensor $s_j \in \mathcal{S}$ by combining the personal evaluations $p_{ij}$ from all possible clients. To ensure accuracy, the importance of each evaluation is carefully considered. The final aggregated reputation for sensor $s_j$, denoted as $as_j$, is determined using the following formula:

$$as_j = \sum_i p_{ij} \cdot \frac{\max(H - (T - t_{ij})), 0)}{H}, \forall i. \tag{2}$$

Equation 2 illustrates that the aggregated sensor reputation is calculated as a weighted sum of all contributed evaluations. To enhance the aggregation process, we incorporate the concept of reputation attenuation, which accounts for the temporal relevance of evaluations. The closer an evaluation is to the current time, the more weight it carries, reflecting its greater significance.

Reputation attenuation entails the proportional decay of reputation values over time. Inspired by the attenuation methods discussed in [38], [39], the weight of $p_{ij}$ is expressed as $\frac{\max(H-(T-t_{ij})),0)}{H}$, where $T$ represents the height of the latest block, and $H$ is a constant defining the acceptable range for the earliest evaluation based on block height. The final aggregated sensor reputation, $as_j$, is computed by summing the weighted contributions from all evaluations made within the recent acceptable time frame.

### B. Aggregated Client Reputation

The aggregated reputation $ac_i$ of client $c_i$ is derived from the aggregated sensor reputations $as_{ij}$. Since the aggregated sensor reputation has been processed, the aggregated client reputation of a client can be calculated based on its bonded sensors. We can use a simple average method to calculate this step,

$$ac_i = \frac{\sum_j as_j b_{ij}}{\sum_j b_{ij}}. \tag{3}$$

The function above illustrates that the aggregated client reputation is derived from the aggregated sensor reputations. This means that a client's reputation depends on the quality of service delivered by all the sensors it is associated with. This approach incentivizes clients to prioritize reliable sensors that enhance service quality while eliminating unreliable sensors to optimize resource utilization.

## V. Sharding Reputation Management

We have introduced the reputation mechanism in Section IV. However, as the number of nodes increases, the computational and storage costs associated with the reputation mechanism can become significant. To address this issue, we employ the sharding technique in this section to mitigate these costs and improve efficiency.

## A. *Resource Cost Issue of Reputation Mechanism*

In Section IV, we introduced a reputation mechanism that enables clients to evaluate each other. However, as the number of clients and sensors increases, the computational and storage costs for managing the reputation system also rise. The total size of the reputation data is $C \times S$, where $C$ is the number of clients and $S$ is the number of sensors, necessitating a corresponding amount of storage for all reputation information. This leads to significant demands on on-chain storage space and network bandwidth. To address these challenges, we implement a sharding mechanism that reduces the computational and storage costs associated with managing reputation data.

Sharding is a technique designed to enhance scalability in blockchain systems by partitioning nodes into multiple groups. Typically, blockchain systems require all nodes to process and validate every transaction across the network, which incurs substantial resource costs. To address this, we employ sharding, where nodes are organized into different committees, each responsible for handling a subset of transactions. In this paper, we use the terms "shard" and "committee" interchangeably.

## B. *Committee Structure*

We split $C$ clients into $M + 1$ committees, including $M$ common committees and a referee committee. Each committee includes multiple clients. Each client belongs to a committee, either the common committee or the referee committee. The member clients of each committee are chosen randomly by various methods, such as the cryptographic sortition in Algorand [40].

*1) Common Committee:* Clients within the same committee continue their usual operations, including collecting data from sensors, requesting data from cloud storage, and evaluating other sensors. The core concept of a committee is to confine evaluations to within the committee if both evaluating and evaluated clients belong to the same committee, thus reducing the need to propagate evaluations across other committees. Only evaluations involving clients from different committees require periodic cross-shard processing. However, when evaluating sensors where the evaluating client is in a different committee, the evaluation cannot be resolved within the same shard. This cross-shard evaluation process will be addressed later.

Each committee, except the referee committee, has a designated leader. The leader is responsible for collecting evaluations from other clients within the same committee. Over a set period, the committee aggregates these evaluations and updates the reputations. Additionally, the leader coordinates with leaders from other committees to share and update reputation information across different committees.

To avoid disconnection or illegal operations from the common committee, all clients in the same common committee have the right to monitor the leader. Clients in the same common committee are responsible for reporting the observed abnormal situations of the leader, thereby keeping the common committee run in expectation. The reports are submitted to the referee committee, which will be discussed in the following content.

*2) Referee Committee:* The referee committee is tasked with handling reports about leaders from the common committees and making judgments based on these reports. When a report is received, the committee members vote, and the majority opinion determines the committee's stance.

If the majority view indicates that a leader has acted improperly, the leader's reputation will be adjusted accordingly. The leader position in the affected common committee will then be reassigned to another client with the highest aggregated reputation within that committee. The referee committee will notify the entire network of this leadership change to ensure continuity in cross-shard transactions.

If the referee committee disagrees with the report, the reputation of the reporting client will be adjusted, and any further reports from that client will be disregarded for the remainder of the current round. This measure helps prevent abuse of the reporting system and protects against potential DDoS attacks.

*3) Weighted Reputation of Clients:* There remains a problem that a client may have honest sensors, but behave dishonestly while maintaining the committee information related to reputations. Since the aggregated reputation $ac_i$ of a client $c_i$ is derived from the aggregated sensor reputations $as_{ij}$, we need another variable $l_i$ to indicate the behavior while $c_i$ is a leader. Initially, all clients $c_i$ have the same $l_i$. If $c_i$ finishes the leader duty during its leader term without being voted out, $l_i$ will increase, and vice versa. The adjustment criterion $l_i$ is the same for different clients because we can only judge the fact that a leader has been voted out, but we cannot judge the extent of the damage. Since the referee committee knows the changes of leaders, $l_i$ is public information and can only be adjusted by the referee committee.

Considering the effectiveness of $l_i$, we need a weighted reputation $r_i$ for each client $c_i$. $r_i$ combines $l_i$ and $ac_i$ as follows:

$$r_i = ac_i + \alpha l_i. \tag{4}$$

$\alpha$ in equation 4 is a constant set by the network, indicating the effectiveness of $l_i$.

## C. *Sharding Reputation Maintainance*

In the design in Section IV, the aggregated reputations are computed by all clients in the network. To reduce computational and storage costs, we adjust the raters of a sensor from other clients to shards. In other words, the reputation of a sensor is aggregated by the views of committees.

Next, we discuss how a committee generates the reputation of a node. The evaluation within each committee is aggregated by the clients who belong to that committee. A simple approach would be for each member of the committee to immediately propose their evaluations to the entire network. However, this method would not reduce the computational and storage costs, as the total number of evaluations across the network remains unchanged. Since the evaluation of sensors is not time-sensitive, periodically updating the aggregated evaluations is an acceptable strategy. This periodic update method helps optimize the use of network resources while maintaining the accuracy of the reputation system.

Equations 2 and 3 are linear, which allows for a straightforward computation of the aggregated sensor reputation $as_j$ and the aggregated client reputation $ac_i$ using information from different committees. The method for aggregating reputations within a single committee, as well as across different committees, is relatively simple. Specifically, the leader of each committee can compute the weighted sum of the aggregated personal reputations for sensors outside the committee, based on the evaluations provided by the clients within that committee. Afterward, all committee leaders use the aggregated results from their respective committees to calculate the overall aggregated reputation of the sensors in their committee.

After computing the cross-shard aggregated reputations, the referee committee is responsible for verifying the accuracy of the results. The process followed by the referee committee mirrors that of the leader's evaluation, where it reviews the leader's actions and ensures the final results are accurate and consistent. This oversight ensures the integrity and correctness of the aggregated reputations across the system.

## D. Off-chain Evaluation Maintainance

Evaluations arise from operations where a client requests data collected by sensors, and theoretically, evidence of these operations should be recorded. However, most clients in the network are not concerned with the evaluation process itself; the primary purpose of these records is for backtracking. Typically, the referee committee will query these off-chain records only when tracing the origin of an evaluation to verify the legality of a client's behavior. Consequently, clients do not generally need to access this information, nor is it necessary to maintain these operations on-chain. To address this, we implement off-chain smart contracts to minimize the number of evaluations that need to be recorded and spread across the network.

The objectives of the off-chain smart contract are as follows: First, it manages evaluations within a shard off-chain to conserve on-chain resources, including network bandwidth and storage. Second, it ensures the collection and protection of evaluations conducted by nodes within a shard. The smart contract is responsible for computing aggregate evaluations and preventing tampering by malicious parties. Third, it ensures that all nodes in a shard reach a consensus on the evaluations. Each node can verify the results and provide signatures if they agree with the outcome presented by the smart contract. While previous work [41]–[43] has addressed the precise implementation of off-chain smart contracts, we focus on the high-level design and functionalities in our discussion.

To maintain reputation accurately, updates should occur periodically. We assume that all nodes within a shard sign up and execute a smart contract once the shard's confirmation is recorded on the blockchain. Upon completion of the smart contract, the updated reputation evaluations of the nodes will be proposed on-chain as verified modifications. If there are changes in the shard's member nodes, the nodes within the same shard will establish a new smart contract. Conversely, if the membership remains the same, the existing nodes will set up a new smart contract. Only one smart contract is executed per shard at any given time.

## E. Performance Analysis

The reputation of all nodes can reach a consensus if the sharding proposal and the last smart contract of each committee are confirmed. Assuming that there are total $C$ clients in the network and $M$ common committees, the number of raters for a sensor is reduced from $C$ to $M$.

The sharding reputation management saves the on-chain storage costs. With the support of off-chain smart contracts, the evaluations are transferred from on-chain storage to the off-chain protocol. The smart contract deals with the evaluations in the shard and outputs the final adjusted reputation of each node based on the off-chain evaluations. Assume that a sensor averagely receives $Q$ evaluations during the lifetime of the smart contract, the number of on-chain evaluations can be reduced from $QS + CS$ to $MS$. It implies that the sharding reputation supports better performances when clients are giving evaluations of sensors frequently. The more frequently a sensor is accessed, the more space is saved.

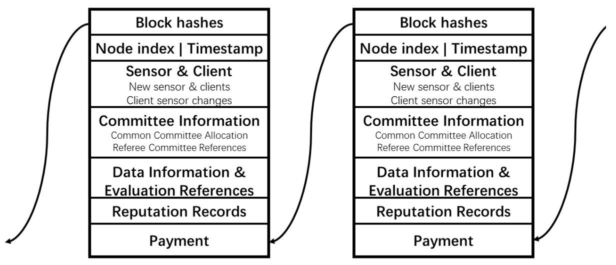
While computing the aggregated client reputation, the number of reputation data pieces is $CS$ originally. With the committee mechanism, each leader computes an intra-shard aggregated client reputation by the weighted sum of $CS/M$ terms. Based on the results from leaders, the final aggregated client reputations can be computed by the weighted sum of $MS$ terms, which reduces the data spread in the whole network.

## VI. Reputation-based Sharding Blockchain

In this section, we introduce a reputation-based sharding blockchain designed to support the proposed sharding reputation mechanism. This blockchain provides a consensus ledger for recording transactions made by nodes in the network and adapts to reputation-based consensus. As illustrated in Figure 2, the blockchain structure integrates the reputation mechanism with sharding reputation management. This structure ensures reliable information recording, allowing clients to reach a consensus on recorded data. The blockchain's immutable ledger is ideal for maintaining accurate records and supporting consensus in edge networks. We then discuss how the blockchain facilitates the recording and management of required information.

## A. General Information

Following the traditional blockchain setup, our blockchain includes block hashes, node indices, timestamps, and payment information. Block hashes, node indices, and timestamps help participants determine the order of blocks and verify their legality. The payment section records transactions such as



(a)

Fig. 2. An overview of blocks applied to our proposed blockchain structure.

payments to cloud storage providers or payments from one client to another for specific data requests. Detailed methods of payment are outside the scope of this paper.

## B. Sensor and Client Information

The sensor and client information section details changes within the network. Specifically, each block records updates related to sensors and clients, as reported by clients during the block's time period. We outline these specific changes.

Clients can propose modifications to their bonded sensors, including adding new sensors or removing existing ones. A new client joining the network will include information about its newly bonded sensors. Existing clients can also update their sensor list to enhance their reputation. If a client wishes to re-use a previously bonded sensor, it must register the sensor under a new identity.

Due to network latency, sensor-client relationships are not immediately updated. All clients apply these changes after the current block has been proposed to the network. Consequently, clients will use sensor and client information from the preceding block.

## C. Committee Information

Aggregated client reputations are computed using Equation 3. Consensus on these aggregated reputations is required from all nodes. The committee structure aids in this consensus process, with the referee committee guiding the judgment. Random selection of the referee committee is proven to be secure, ensuring that with high probability, more than half of the clients are honest [44]. Specifically, if the expected committee size is $\Theta(\log^2 S)$, the probability of an insecure committee is less than $n^{\frac{-\log n}{12}}$, which is negligible compared to $S$.

Each block records the committee membership of all clients. For each common committee, the block details its members and leader, which are later compared against reputation records. Additionally, the block includes records of the referee committee members, who address potential issues in the common committees. Voting records and electronic signatures of each client report are also recorded for reference. The system provides rewards to the leader and members of the referee committee in the payment section to incentivize their contributions to the current block. This reward mechanism forms the basis for their compensation.

## D. Data Information and Evaluation References

The data collected by sensors is managed by the corresponding client. Each client gathers raw data from multiple sensors, processes or refines it as needed, and then uploads the data to cloud storage. The client subsequently makes the information about the uploaded data available to other clients for potential use. When another client requests this data, its quality can be assessed based on its evaluations.

As described in Section V, evaluations by clients are not recorded on-chain. Instead, off-chain evaluations are supported by smart contracts. Committee leaders store the relevant smart contract information in cloud storage, and the addresses of this information are recorded on the blockchain for reference. If a client needs to retrieve historical information for a specific evaluation, it can consult the blockchain and request the necessary details from cloud storage. Payments related to these transactions are handled directly between the client and the cloud storage provider, so there is no need to record them on-chain.

## E. Proof-of-Reputation

The primary goal of a consensus mechanism is to establish temporary trust within an untrusted network environment. PoR leverages reputation as a key metric, offering advantages such as low energy consumption and robust security. Unlike PoW which demands significant energy resources, PoR operates with minimal energy requirements. Its security is rooted in the use of reputation, ensuring that trusted and reputable clients are incentivized to act honestly.

The core principle of PoR is to entrust the maintenance of the blockchain to clients with the highest reputations. These high-reputation clients earn trust by consistently acting with integrity and maintaining reliable sensors, fostering a trustworthy and efficient network.

The reputation metric used in PoR is the weighted reputation, denoted as $r_i$. As a consensus mechanism, PoR determines which clients are responsible for block maintenance. Within each committee, the client with the highest $r_i$ is automatically designated as the leader. In the context of a sharding blockchain, the leader of each committee performs two key functions: aggregating sensor reputation ($as_j$) and client reputation ($ac_i$). Once the aggregation is complete, the leader interacts with the referee committee to upload both intra-shard and cross-shard reputations, ensuring consistency and coordination across the blockchain.

If the current leader in a committee is reported as malicious by the majority of its members, the referee committee will appoint a new leader. This new leader is selected from the remaining unreported members of the committee, with the client having the highest $r_i$ taking the leader.

### F. Generating Blocks and Reputation Records

When the number of data exchange transactions and reputation updates reaches a certain threshold or period, a new block is necessary to package these updates and record them on the blockchain. Leaders begin by exchanging aggregated reputations and evaluation references for sensors and clients within the committee. They then compute the updated reputations, vote on them, and submit proposals to the referee committee for final review. The referee committee performs a final assessment, and if more than half of the leaders and referees approve, the new block is generated and broadcast to the entire network. The block also includes the updated committee allocations, calculated using the algorithm from Gilad et al. [40].

Blocks must accurately record the most recent reputation information. Specifically, the generators of the current block calculate updated aggregated sensor and client reputations and include these in the block. This process is overseen by the referee committee as part of the consensus mechanism. Clients use the reputations computed and recorded in the current block as needed and update their reputation records accordingly until the next block is accepted. Consequently, all nodes adhere to the up-to-date reputations recorded in the latest block.

The client with the highest aggregated reputation within the current committee is selected as the leader. Given that interactions between different shards rely on leaders with high aggregated reputations, an additional key responsibility of the leader is to generate new blocks. This selection process aims to be as fair as possible to ensure validation from referee nodes.

## VII. Performance Evaluation

This section presents the performance evaluation of our implemented reputation-based sharding blockchain system. We begin by defining a standard test setting. Using this setting, we adjust key variables to analyze their impact on system effectiveness. The simulation results illustrate the system's performance across multiple dimensions, including on-chain data size, service quality, and reputation maintenance.

### A. Standard Test Setting

Without special clarification, the simulations we show in this section have the following settings. The network has 10,000 sensors and 500 clients by default. The clients are divided into 10 committees. Clients can access data from any sensor, with the data being influenced by the sensor's data quality. The data quality is set as 0.9 for all sensors, indicating that with 0.9 probability a sensor provides good data, and 0.1 probability provides bad data.

We run each simulation until the number of blocks reaches 1000. During the interval between two block generations, we randomly perform 1000 operations, including:

- Sensor Data Generation: A sensor $s_j$ generates new data, with the 0.9 probability data is good, and 0.1 probability data is bad.
- Data Access and Evaluation: A client $c_i$ accesses existing data, and updates the sensor's reputation based on this evaluation.

Since the global reputation system is standardized, clients can implement any reputation mechanism within the framework. For simplicity, our simulation assumes all clients use the same standardized reputation formula: $p_{ij} = pos_{ij}/tot_{ij}$, where $p_{ij}$ represents the personal reputation between client $c_i$ and sensor $s_j$. Here, $pos_{ij}$ is the count of positive data accesses, and $tot_{ij}$ is the total number of data accesses. Initially, $pos_{ij} = tot_{ij} = 1$.

When calculating the aggregated sensor reputation, the constant $H$ in Equation 2 is set to 10, and the default value of $\alpha$ in Equation 4 is 0. To avoid accessing unreliable data, client $c_i$ only interacts with sensors $s_j$ that satisfy $p_{ij} \geq 0.5$. Similarly, $l_i$ is computed using the same approach as $p_{ij}$, representing the ratio of successful completions of leader duties to the total number of times the leader duty is performed.

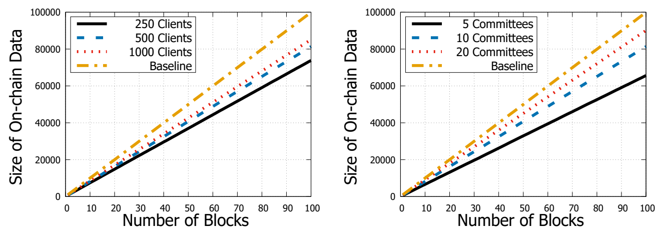
### B. On-chain Data Size Test

In previous studies, transactions per second (TPS) and latency are commonly used metrics to evaluate blockchain efficiency. However, these metrics are highly dependent on environmental factors such as network bandwidth and the computational power of the clients involved. To provide a more stable and unbiased measurement, we focus on the amount of on-chain data as an alternative metric. Unlike TPS or latency, the size of on-chain data is not influenced by the aforementioned environmental variables. Given the rapid growth of on-chain data size, we limit our results to the first 100 blocks for clarity and relevance.

The on-chain data size test compares the amount of on-chain data between our proposed structure and a baseline. The baseline follows the same reputation behavior but with different on-chain storage rules, where all evaluations are uploaded to the main chain and recorded. To assess the scalability of the system, we evaluate the performance with different numbers of clients and committees. While the data size grows linearly, the results show that our proposed structure consistently stores less data compared to the baseline, which lacks committee optimization.

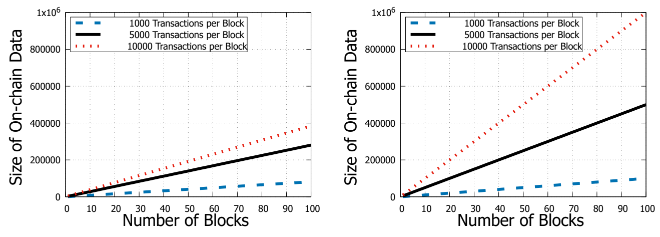
Specifically, Fig. 3 illustrates the performance of both methods. In Fig. 3(a), we vary the number of clients in the network, including 250, 500, and 1000 clients. Notably, the baseline results remain unchanged regardless of the number of clients or committees, as the total number of evaluations remains constant. Our proposed system performs better when the number of clients is low, as sensors are more likely to be associated with a single client, and evaluations from one committee can evaluate fewer clients from other committees.

In Fig. 3(b), we vary the number of committees, testing values of 5, 10, and 20. As the number of committees decreases, the size of on-chain data also reduces because the number of members in each committee increases, resulting in fewer intra-shard evaluations occupying on-chain storage. However, this result does not imply that fewer committees are always



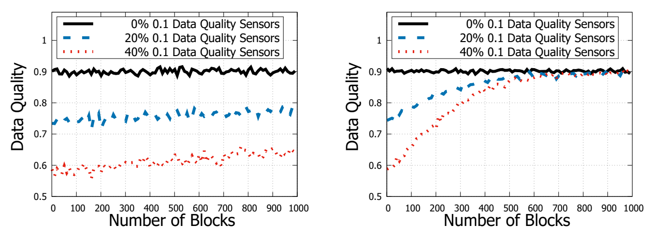
(a) (b)

Fig. 3. The size of on-chain data across different networks. (a) shows the results for varying numbers of clients, and (b) shows the results for different numbers of committees.



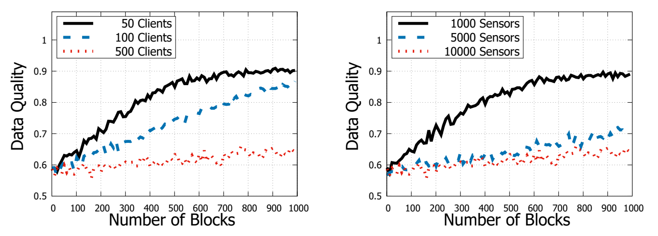
(a) (b)

Fig. 4. The size of on-chain data under different numbers of evaluations during the block generation period. (a) shows the results computed using our proposed sharding mechanism, while (b) shows the results based on the baseline.



(a) (b)

Fig. 5. The data quality changes over time. (a) shows the results when the number of evaluations per block is 1000. (b) shows the results when the number of evaluations per block is 5000.



(a) (b)

Fig. 6. The change in data quality over time when 40% of sensors have a data quality of 0.1. (a) shows the results for different numbers of clients, while (b) shows the results for varying numbers of sensors.

better. Reducing the number of committees places additional pressure on the leaders of each committee and can degrade the efficiency of voting and consensus processes within the committee.

The effectiveness of the sharding technique increases as the number of evaluations during the block generation period rises. Fig. 4 shows the results of adjusting the number of evaluations per block. Fig. 4(a) presents the results from our proposed sharding mechanism, while Fig. 4(b) shows the baseline results. While there is little difference observed when the number of evaluations per block is 1000, the reduction in on-chain data size becomes more significant as the evaluations per block increase to 5000 and 10000. Specifically, when the number of blocks reaches 100, our proposed sharding mechanism reduces the size of on-chain data to 85.13%, 56.07%, and 38.36% of the baseline for 1000, 5000, and 10000 evaluations per block, respectively. The longer the blockchain operates, the more storage space is saved on-chain.

### *C. Service Quality Test*

Ensuring service quality is another key objective of our proposed system. To evaluate its performance, we adjust the service quality of the sensors and present the results. Specifically, we introduce a proportion of sensors with poor quality, where only 10% of the data provided is good, while 90% of the data is bad.

Fig. 5 presents the results when 0%, 20%, and 40% of sensors have poor quality. Following the standard test setting, Fig. 5(a) tracks the data quality of each block. The data quality aligns with the initial expectations of 0.9, 0.74, and 0.58, corresponding to 0%, 20%, and 40% of bad quality sensors, respectively. As the number of blocks increases, data quality improves because sensors with low quality (0.1) are filtered out. However, the result in Fig. 5(a) is not very pronounced. To accelerate this improvement, we increase the number of evaluations per block, from 1000 to 5000, and the results are shown in Fig. 5(b). In this case, both the 20% and 40% bad quality sensor scenarios reach a data quality of 0.9 when the number of blocks approaches 650. These simulations demonstrate that our proposed reputation-based sharding system effectively helps clients improve data quality and identify unreliable sensors.

We then adjust the network size to observe the convergence speed of data quality. The simulations are based on the scenario where 40% of sensors provide a data quality of 0.1, making it easier to observe the phenomenon. As we adjust the number of clients and sensors, the convergence speed of data quality also changes. Fig. 6(a) shows the results for different client counts (50, 100, 500). When there are 50 clients, the data quality converges to 0.9 after 700 blocks. With 100 clients, the data quality stabilizes around 0.86 after 1000 blocks. In Fig. 6(b), when the number of sensors is reduced to 1000, the convergence trend is similar to the 50-client case, with the data quality reaching 0.9 at 700 blocks. With 5000 sensors, the data quality converges to around 0.7. From these results, we observe that the convergence speed is related to the product

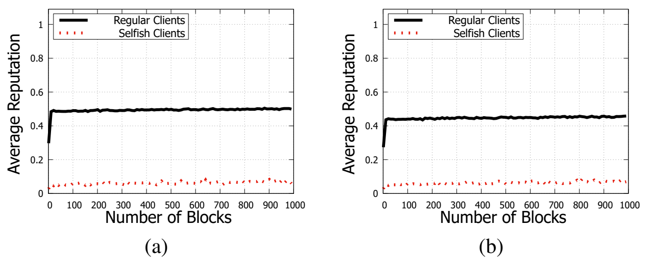


Fig. 7. The change of average reputation of clients when a proportion of clients are selfish. (a) shows the results for 10% of selfish clients, while (b) shows the results for 20% of selfish clients.

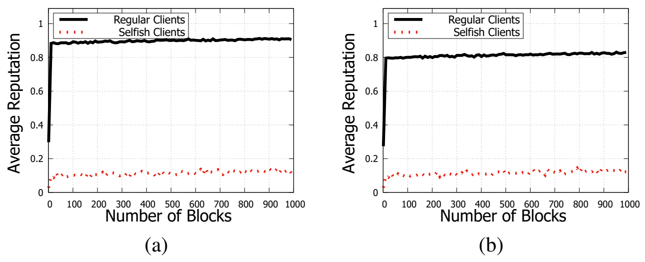


Fig. 8. The change of average reputation of clients when a proportion of clients are selfish. In these simulations, the attenuation mechanism is not applied. (a) shows the results for 10% of selfish clients, while (b) shows the results for 20% of selfish clients.

of the number of clients and sensors.

### D. Reputation of Clients

We design simulations to evaluate client reputations by categorizing clients into regular and selfish types. The sensors of regular clients provide data with a quality of 0.9, while the sensors of selfish clients give 0.9 quality data to selfish clients but only 0.1 quality data to regular clients. We track the aggregated client reputations of both types. In Fig. 7(a), 10% of the clients are selfish, and in Fig. 7(b), 20% of the clients are selfish.

Two key observations emerge from the simulations. First, in both figures, the reputations of regular and selfish clients stabilize quickly and then rise gradually. This behavior indicates that our reputation system effectively reflects client behavior through their bonded sensors. Second, the reputation of selfish clients stabilizes around 0.06 in both figures, while the reputation of regular clients converges to 0.49 in Fig. 7(a) and 0.44 in Fig. 7(b). These results suggest that, although selfish clients impact the reputations of regular clients, regular clients maintain a significant advantage in reputation as long as the proportion of selfish clients remains below a threshold.

A problem persists where the reputations of regular and selfish clients appear to be about half of our expected values. This issue arises because the reputation value is affected by the attenuation mechanism (as described in equation 2). Since interactions between specific clients and sensors may not occur frequently in the simulation, the attenuation mechanism causes a decrease in the reputation value over time. Fig. 8(a) represents the version of Fig. 7(a) without attenuation, and Fig. 8(b) corresponds to the attenuation-free version of Fig. 7(b). In these cases, the expected reputation values are 0.1 for selfish clients and 0.9 for regular clients. In Fig. 8(a), the calculated reputation values align with expectations, showing that the system behaves as anticipated. In Fig. 8(b), however, the reputation of selfish clients maliciously drags the average reputation down to around 0.8, which is consistent with the results observed in Fig. 7(b). This illustrates that the attenuation mechanism plays a significant role in the final reputation values over time.

## VIII. Conclusion

In this paper, we have introduced a novel approach for managing sensor networks in edge computing environments through the design of a reputation mechanism integrated with a sharding blockchain. Our reputation mechanism enables clients to evaluate and share the reputations of sensors, which in turn affects the reputation of the clients managing those sensors. By leveraging this mechanism, we developed a reputation-based sharding blockchain that addresses key challenges associated with scalability and efficiency in large sensor networks.

The proposed sharding blockchain not only enhances scalability but also ensures that the system can support a substantial number of clients and sensors. This structure allows for efficient management of sensor data and reputation, facilitates data requests, and maintains consensus across the blockchain with minimal additional resource consumption. The extensive simulations conducted validate the effectiveness of our approach, demonstrating that it can handle large-scale sensor networks while maintaining robust network consensus.

Overall, our work provides a scalable and efficient solution for reputation management in edge networks, offering a significant advancement in blockchain technology and its application to decentralized sensor systems. Future research may focus on further optimizing the reputation mechanism and sharding process, exploring additional use cases, and evaluating the system's performance in more diverse and real-world scenarios.

## Acknowledgement

This work is supported in part by US National Science Foundation under grant number 2230620.

## References

[1] T. Wang, Y. Liang, X. Shen, X. Zheng, A. Mahmood, and Q. Z. Sheng, "Edge computing and sensor-cloud: Overview, solutions, and directions," *ACM Computing Surveys*, vol. 55, no. 13, pp. 1–37, 2023.

[2] J. P. Bharadiya, "The role of machine learning in transforming business intelligence," *International Journal of Computing and Artificial Intelligence*, vol. 4, no. 1, pp. 16–24, 2023.

[3] O. T. Modupe, A. A. Otitoola, O. J. Oladapo, O. O. Abiona, O. C. Oyeniran, A. O. Adewusi, A. M. Komolafe, and A. Obijuru, "Reviewing the transformational impact of edge computing on real-time data processing and analytics," *Computer Science & IT Research Journal*, vol. 5, no. 3, pp. 693–702, 2024.

[4] J. S. Raj and S. Jennifer, “Optimized mobile edge computing framework for iot based medical sensor network nodes,” *Journal of Ubiquitous Computing and Communication Technologies (UCCT)*, vol. 3, no. 01, pp. 33–42, 2021.

[5] M. A. Khan, E. Baccour, Z. Chkirbene, A. Erbad, R. Hamila, M. Hamdi, and M. Gabbouj, “A survey on mobile edge computing for video streaming: Opportunities and challenges,” *IEEE Access*, vol. 10, pp. 120 514–120 550, 2022.

[6] Y. Shang, Z. Li, S. Li, Z. Shao, and L. Jian, “An information security solution for vehicle-to-grid scheduling by distributed edge computing and federated deep learning,” *IEEE Transactions on Industry Applications*, 2024.

[7] A. Nawaz, J. Peña Queralta, J. Guan, M. Awais, T. N. Gia, A. K. Bashir, H. Kan, and T. Westerlund, “Edge computing to secure iot data ownership and trade with the ethereum blockchain,” *Sensors*, vol. 20, no. 14, pp. 3965–3965, 2020.

[8] S. Nakamoto *et al.*, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[9] M. I. Khalid, I. Ehsan, A. K. Al-Ani, J. Iqbal, S. Hussain, S. S. Ullah *et al.*, “A comprehensive survey on blockchain-based decentralized storage networks,” *IEEE Access*, vol. 11, pp. 10 995–11 015, 2023.

[10] X. Zhang, J. Grannis, I. Baggili, and N. L. Beebe, “Frameup: an incriminatory attack on storj: a peer to peer blockchain enabled distributed storage system,” *Digital Investigation*, vol. 29, pp. 28–42, 2019.

[11] D. Vorick and L. Champine, “Sia: Simple decentralized storage,” *Retrieved May*, vol. 8, pp. 2018–2018, 2014.

[12] Y. Psaras and D. Dias, “The interplanetary file system and the filecoin network,” in *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*. IEEE, 2020, pp. 80–80.

[13] P. Kochovski, S. Gec, V. Stankovski, M. Bajec, and P. D. Drobintsev, “Trust management in a blockchain based fog computing platform with trustless smart oracles,” *Future Generation Computer Systems*, vol. 101, pp. 747–759, 2019.

[14] H. Xu, D. Liu, H. Wang, and A. Stavrou, “E-commerce reputation manipulation: The emergence of reputation-escalation-as-a-service,” in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1296–1306.

[15] M. Srivatsa, L. Xiong, and L. Liu, “Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks,” in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 422–431.

[16] P. Weerapanpisit, S. Trilles, J. Huerta, and M. Painho, “A decentralized location-based reputation management system in the iot using blockchain,” *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 15 100–15 115, 2022.

[17] H. Xie, Y. Li, and J. C. Lui, “Optimizing discount and reputation tradeoffs in e-commerce systems: Characterization and online learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 7992–7999.

[18] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, “Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach,” in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[19] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, “Resource trading in blockchain-based industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3602–3609, 2019.

[20] W. Sun, J. Liu, Y. Yue, and P. Wang, “Joint resource allocation and incentive design for blockchain-based mobile edge computing,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 6050–6064, 2020.

[21] Y. Yuan and F.-Y. Wang, “Towards blockchain-based intelligent transportation systems,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 2663–2668.

[22] G. Ateniese, L. Chen, M. Etemad, and Q. Tang, “Proof of storage-time: Efficiently checking continuous data availability,” *Cryptology ePrint Archive*, 2020.

[23] C. Zhang, X. Li, and M. H. Au, “epost: Practical and client-friendly proof of storage-time,” *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1052–1063, 2023.

[24] F. Gai, B. Wang, W. Deng, and W. Peng, “Proof of reputation: A reputation-based consensus protocol for peer-to-peer network,” in *International Conference on Database Systems for Advanced Applications*. Springer, 2018, pp. 666–681.

[25] J. Yu, D. Kozhaya, J. Decouchant, and P. Esteves-Verissimo, “Repucoin: Your reputation is your power,” *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1225–1237, 2019.

[26] M. T. de Oliveira, L. H. Reis, D. S. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. Mattos, “Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications,” *Computer Networks*, vol. 179, pp. 107 367–107 367, 2020.

[27] E. K. Wang, Z. Liang, C.-M. Chen, S. Kumari, and M. K. Khan, “Porx: A reputation incentive scheme for blockchain consensus of iiot,” *Future Generation Computer Systems*, vol. 102, pp. 140–151, 2020.

[28] G. Qiao, S. Leng, H. Chai, A. Asadi, and Y. Zhang, “Blockchain empowered resource trading in mobile edge computing and networks,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[29] Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye, and Y. Yang, “Resource allocation and consensus on edge blockchain in pervasive edge computing environments,” in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1476–1486.

[30] X. Lin, J. Wu, S. Mumtaz, S. Garg, J. Li, and M. Guizani, “Blockchain-based on-demand computing resource trading in iov-assisted smart city,” *IEEE Transactions on Emerging Topics in Computing*, 2020.

[31] F. Ayaz, Z. Sheng, D. Tian, and Y. L. Guan, “A proof-of-quality-factor (poqf) based blockchain and edge computing for vehicular message dissemination,” *IEEE Internet of Things Journal*, 2020.

[32] X. Huang, R. Yu, J. Kang, and Y. Zhang, “Distributed reputation management for secure and efficient vehicular edge computing and networks,” *IEEE Access*, vol. 5, pp. 25 408–25 420, 2017.

[33] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, “Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 695–708, 2018.

[34] L. Xiao, Y. Ding, D. Jiang, J. Huang, D. Wang, J. Li, and H. V. Poor, “A reinforcement learning and blockchain-based trust mechanism for edge networks,” *IEEE Transactions on Communications*, 2020.

[35] Y. Singh, F. Kandah, and W. Zhang, “A secured cost-effective multi-cloud storage in cloud computing,” in *2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 2011, pp. 619–624.

[36] B. Guidi, A. Michienzi, and L. Ricci, “Evaluating the decentralisation of filecoin,” in *Proceedings of the 3rd International Workshop on Distributed Infrastructure for the Common Good*, 2022, pp. 13–18.

[37] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 640–651.

[38] J. Zhang, Y. Cheng, X. Deng, B. Wang, J. Xie, Y. Yang, and M. Zhang, “A reputation-based mechanism for transaction processing in blockchain systems,” *IEEE Transactions on Computers*, vol. 71, no. 10, pp. 2423–2434, 2021.

[39] S. Buchegger and J.-Y. Le Boudec, “A robust reputation system for mobile ad-hoc networks,” Technical Report IC/2003/50, EPFL-IC-LCA, 2003.

[40] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, “Algorand: Scaling byzantine agreements for cryptocurrencies,” in *Proceedings of the 26th symposium on operating systems principles*, 2017, pp. 51–68.

[41] D. Cacciagrano, F. Corradini, G. Mazzante, L. Mostarda, and D. Sestili, “Off-chain execution of iot smart contracts,” in *International Conference on Advanced Information Networking and Applications*. Springer, 2021, pp. 608–619.

[42] J. Eberhardt and S. Tai, “Zokrates-scalable privacy-preserving off-chain computations,” in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1084–1091.

[43] C. Li, B. Palanisamy, and R. Xu, “Scalable and privacy-preserving design of on/off-chain smart contracts,” in *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2019, pp. 7–12.

[44] M. Zhang, J. Li, Z. Chen, H. Chen, and X. Deng, “An efficient and robust committee structure for sharding blockchain,” *IEEE Transactions on Cloud Computing*, 2022.