# SEMSO: A Secure and Efficient Multi-Data Source Blockchain Oracle

Youquan Xian, Xueying Zeng, Chunpei Li, Peng Wang, Dongcheng Li, Peng Liu, and Xianxian Li

***Abstract*—In recent years, blockchain oracle, as the key link between blockchain and real-world data interaction, has greatly expanded the application scope of blockchain. In particular, the emergence of the Multi-Data Source (MDS) oracle has greatly improved the reliability of the oracle in the case of untrustworthy data sources. However, the current MDS oracle scheme requires nodes to obtain data redundantly from multiple data sources to guarantee data reliability, which greatly increases the resource overhead and response time of the system. Therefore, in this paper, we propose a Secure and Efficient Multi-data Source Oracle framework (SEMSO), where nodes only need to access one data source to ensure the reliability of final data. First, we design a new off-chain data aggregation protocol TBLS, to guarantee data source diversity and reliability at low cost. Second, according to the rational man assumption, the data source selection task of nodes is modeled and solved based on the Bayesian game under incomplete information to maximize the node's revenue while improving the success rate of TBLS aggregation and system response speed. Security analysis verifies the reliability of the proposed scheme, and experiments show that under the same environmental assumptions, SEMSO takes into account data diversity while reducing the response time by 23.5%.**

***Index Terms*—Blockchain, blockchain oracles, multi-data source (MDS), reinforcement learning, Bayesian game.**

## I. Introduction

In RECENT years, with the rapid development of blockchain technology, its applications have become increasingly widespread in areas such as Decentralized Finance (DeFi) [1], supply chain management [2], and healthcare [3]. However, the closed nature of blockchain networks makes it difficult to directly access and process real-world data, limiting their potential and scope in practical applications. To address this limitation, blockchain oracles have emerged as bridges for data interaction between on-chain and off-chain environments. It is responsible for retrieving, verifying, and submitting external data to the blockchain, significantly expanding its application scenarios and development potential [4].

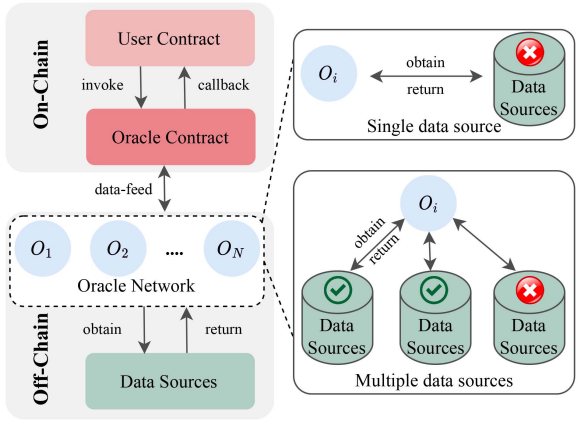



Fig. 1. SDS and MDS oracle.

In the early days, oracle schemes usually used voting games [5], [6], threshold signature [7], [8], reputation mechanisms [9], [10], etc. methods to construct a distributed oracle node trust system, or used a Trusted Execution Environment (TEE) [11], [12] and improved protocol of TLS (such as TLS-N [13]) to ensure the reliability of data acquisition. However, it is essential to acknowledge that data sources may also be untrustworthy. Blindly trusting data from a Single Data Source (SDS) can lead to significant financial losses[1] [14], [15].

While Multi-Data Sources (MDS) oracle solutions enhance data reliability by establishing a distributed trust system through the diversification of data sources, they also introduce significant challenges [16], [17], [18], [19], [20]. Ensuring data source diversity necessitates that individual nodes access multiple sources, significantly increasing system resource consumption, operational costs, and response times. This challenge becomes even more pronounced in scenarios where data access incurs high costs, ultimately diminishing the willingness of both nodes and users to participate. Such drawbacks not only negatively impact user experience but also impede the practical adoption of oracle technology. Fig. 1 illustrates the fundamental structures and distinctions between SDS and MDS oracles.

Received 16 October 2024; revised 27 May 2025; accepted 1 July 2025. Date of publication 10 July 2025; date of current version 17 October 2025. The work was supported in part by the Guangxi Science and Technology Major Project under Grant AA22068070, in part by the National Natural Science Foundation of China under Grant 62166004, Grant U21A20474, Grant 62262003, in part by the Basic Ability Enhancement Program for Young and Middle-aged Teachers of Guangxi under Grant 2022KY0057, Grant 2023KY0062. Recommended for acceptance by J. Carretero. *(Youquan Xian and Xueying Zeng contributed equally to this work.) (Corresponding author: Peng Liu.)*

The authors are with the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China (e-mail: xianyouquan@stu.gxnu.edu.cn; xyz@stu.gxnu.edu.cn; licp@gxnu.edu.cn; wangp@gxnu.edu.cn; ldc@gxnu.edu.cn; liupeng@gxnu.edu.cn; lixx@gxnu.edu.cn).

Digital Object Identifier 10.1109/TPDS.2025.3586450

[1]In 2019, Synthetix, a DeFi project on Ethereum, experienced an incident where a commercial API intermittently reported an exchange rate for the Korean won 1,000 times higher than the actual rate. This erroneous data was adopted by the price feed contract, resulting in financial implications amounting to nearly $1 billion. https://blog.synthetix.io/response-to-oracle-incident/

TABLE I
COMPARISON OF THE PROPOSED SCHEME WITH PREVIOUS APPROACHES

| Scheme | Untrusted Data Sources | Number of Data Accesses | Response Speed | Diversity of Data Sources |
|---|---|---|---|---|
| SDS Schemes | ✘ | $O(N)$ | Fast | ✘ |
| MDS Schemes | ✔ | $O(N \times M)$ | Slow | ✔ |
| Ours | ✔ | $O(N)$ | Fast | ✔ |

Although solutions such as IoT [10] try to allocate fast response nodes to task data sources through reputation mechanisms to improve the response speed. However, it relies on the response time of TEE-trusted record nodes and uses smart contracts to calculate and store reputation values. This leads to a lot of additional overhead, and it is difficult to ensure the diversity of data sources. Therefore, the core challenge is whether it is possible to access a single data source only through nodes without relying on TEE and smart contracts while ensuring the ultimate reliability of the data.

To address the aforementioned problem, this study introduces a secure and efficient multi-source oracle framework SEMSO. It significantly optimizes data access frequency and system response speed while maintaining the diversity of data sources. First, we present a novel off-chain data aggregation protocol, TBLS, which is designed to ensure the final diversity of data sources with minimal resource expenditure. Second, leveraging the TBLS protocol and the assumption of rational agents, we model the nodes using a Bayesian game framework. It aims to maximize node rewards while concurrently enhancing the success rate of TBLS aggregation and improving overall system response times. Table I highlights the distinctions between the proposed framework and previous solutions.

The main contributions of this paper are as follows:

- We designed a data aggregation protocol, TBLS, which integrates TLS-N with threshold signatures to ensure data source diversity and reliability while maintaining low resource expenditure.
- Based on a Bayesian game model, we optimized the data source selection strategy for off-chain nodes, resulting in enhanced data aggregation success rates and improved system response times.
- Both experimental and security analyses validate the reliability of the proposed framework. Notably, the approach not only maintains data diversity but also achieves a 23.5% reduction in response times while significantly decreasing overall system resource consumption.

The remainder of this paper is structured as follows: Section II introduces the relevant background knowledge and existing challenges. Section III offers a brief description of the workflow of the proposed framework SEMSO. Section IV presents the implementation details of the proposed solution. Section V discusses the experimental results and security analysis. Finally, Section VI concludes the paper and outlines future research directions.

## II. RELATED WORK AND EXISTING CHALLENGES

Blockchain oracles serve as a trusted bridge connecting blockchains with the external world, providing blockchains with access to external data that they cannot retrieve autonomously, such as weather conditions, prices, exchange rates, etc. It extends the range of blockchain applications [21].

To tackle the issue of ensuring the trustworthiness of data obtained by oracles, projects like Augur [5] and Astraea [6] introduced mechanisms where multiple oracle nodes vote and engage in game theory, placing bets on the veracity of the data. Deepthought [22] extended Astraea's voting system by linking it to user reputation, incentivizing the most honest users while reducing the risk of corruption caused by adversaries or passive voters, thereby improving data credibility. With the growing popularity of Trusted Execution Environments (TEE), researchers such as Zhang et al. [11], Liu et al. [12], and Woo et al. [23] have combined TEE with oracles to ensure the integrity and consistency of the data being retrieved. In addition, threshold signatures [7], [8], [24] and improved protocols of TLS, such as TLS-N [13], [25], [26], use cryptographic algorithms to ensure the reliability of data, resulting in higher security compared to others. However, all of these approaches inherently assume the trustworthiness of the data sources, making it challenging to ensure their effectiveness when the data sources themselves are untrustworthy.

To mitigate reliance on an SDS, MDS oracles have emerged. By integrating multiple data sources, they enhance data reliability in scenarios where data sources may be untrustworthy. Researchers like Lv et al. [16] and Almi'Ani et al. [17] have employed reputation mechanisms and weighted graphs to quantify the trustworthiness of data sources, thereby recommending reliable sources for tasks. Gigli et al. [18] proposed the DESMO architecture, which assigns reputation scores to data sources to select multiple trustworthy ones for data retrieval. DAON [19] utilizes distributed data sources when the trustworthiness of individual sources is indeterminable, collecting data by querying a set of sources and employing strategies such as majority voting to obtain a single answer. Xiao et al. [20] addressed situations where both oracle nodes and data sources are untrustworthy by using a two-stage truth discovery process to approximate the true data values. Although MDS oracles enhance data reliability by increasing the diversity of data sources and mitigating the negative impact of a few malicious sources, they require oracle nodes to obtain data redundantly from multiple sources to prevent single points of failure or attacks from malicious data sources. This significantly increases resource overhead and response time, reducing participation enthusiasm and system efficiency.

Although solutions such as IoT [10] try to allocate fast response nodes to task data sources through reputation mechanisms to improve the response speed. However, it relies on the response time of TEE-trusted record nodes and uses smart contracts to calculate and store reputation values. This leads to a lot of additional overhead, and it is difficult to ensure the diversity of data sources. Therefore, the core challenge is whether it is possible to access a single data source only through nodes without relying on TEE and smart contracts while ensuring the ultimate reliability of the data.

Many scholars have conducted in-depth research on blockchain oracles. Nevertheless, there remain two critical challenges that urgently need to be addressed:

*a) Challenge 1:* Oracle must develop a novel off-chain data aggregation protocol that can not only guarantee the dependability of data in the form of a threshold signature but also authenticate the data's provenance and guarantee the diversity and reliability of the data.

*b) Challenge 2:* The method of node redundancy, traversing all data sources, is inefficient, and random access to a single data source cannot ensure the diversity of the final data. Therefore, a reasonable data source selection strategy is required to guarantee the diversity and response speed of the final data source when the node only accesses one data source.

## III. SEMSO Overview

The process of an oracle task typically begins with a user contract (e.g., a currency exchange contract). The user contract calls the oracle contract interface to trigger a data request event (e.g., the current exchange rate) and pays a service fee. The oracle contract receives the request task and writes it into the blockchain event. The system consists of $N$ oracle nodes and $M$ data sources. Off-chain oracle nodes $\{\mathcal{O}_1, \ldots, \mathcal{O}_N\}$, abbreviated as nodes $\mathcal{O}_i$, retrieve the required data from data sources $\{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$ (multiple exchange rate APIs), where data sources are denoted as $\mathcal{D}_j$. Different schemes have different request strategies. For example, DAON [19] and DecenTruth [20] etc. MDS schemes require each node to traverse $\{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$ to obtain data. Off-chain data aggregation is commonly used to reduce the cost of data being uploaded to the blockchain, where only the final result is uploaded. Among these methods, threshold signature-based aggregation is the most prevalent, ensuring that only participants meeting a predefined threshold can collaboratively generate a valid signature. This approach not only safeguards private key security but also prevents the freeloading problem, thereby enhancing security and resistance to attacks [27]. The oracle contract then verifies the aggregated result from the oracle network and allocates rewards to the $\mathfrak{t}$ successful nodes to incentivize honest execution of tasks. Finally, the oracle contract calls a callback interface to return the data to the user contract, completing the task.

To address the two key challenges mentioned above, this paper proposes several improvements to off-chain oracle networks. Specifically, we introduce a novel off-chain data aggregation method, TBLS, which balances distributed trust with data source diversity. Additionally, building on TBLS, we enhance the user request strategy such that a single node only needs to access one data source $\mathcal{D}_j$, while still ensuring the diversity of the final aggregated data.

The process flow of the proposed scheme is illustrated in Fig. 2. The detailed steps are as follows:

1) The user contract calls the oracle contract interface to initiate a task request $\mathcal{Q}$, which includes the task ID $\mathcal{I}$, the requested data source set $\mathcal{D}$ and their CA certificate set $\mathcal{C}$. This interaction is performed through the proxy contract within the oracle contract, and tokens are staked and locked in the payment contract. The CA certificate is used to verify the identity of the data source during data access and ensure the reliability of data interactions.

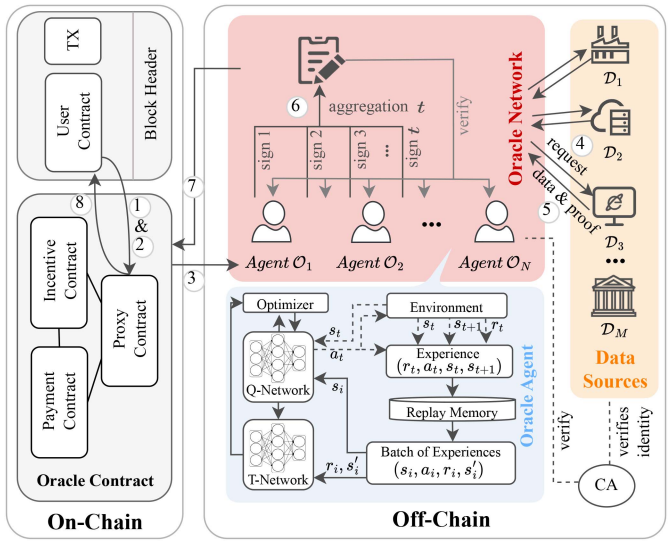



Fig. 2. SEMSO overview.

2) The oracle contract generates an oracle event based on the request event $\mathcal{Q}$, which is recorded on the blockchain.
3) Node $\mathcal{O}_i$ continuously listens for external data request events on the blockchain.
4) Node $\mathcal{O}_i$, acting as an agent in reinforcement learning, uses a DDQN-based reinforcement learning strategy [28] to select the optimal data source $\mathcal{D}_j$ from the requested data source set $\mathcal{D}$, such that the data it reports is most likely to be aggregated by the TBLS, thereby earning rewards. This approach enhances system efficiency while ensuring data source diversity (detailed in Section IV-B).
5) Node $\mathcal{O}_i$ generates proof of data source $\mathcal{P}_i^j$ based on the session and signs the retrieved data $d_j$ with $\sigma_i^j = sign(d_j, sk_i)$.
6) Node $\mathcal{O}_i$ broadcasts and verifies $(\mathcal{I}, \{\mathcal{P}_i^j\}, \sigma_i^j)$ using the TBLS protocol. The final aggregation result is generated based on diversity requirements $K$ and the threshold $\mathfrak{t}$ (detailed in Section IV-A).
7) The first node $\mathcal{O}_i$ to successfully aggregate the result uploads the final aggregated result $\sigma$ and the original data $d$ to the blockchain for verification. The incentive contract then rewards or penalizes the oracle nodes based on their behavior, and the payment contract transfers tokens to the successful oracle nodes as compensation.
8) Finally, the oracle contract calls the callback interface to return the data to the user contract.

*a) Security Assumptions:* Similar to schemes such as DAON [19] and DecenTruth [20], the system consists of $N$ oracle nodes and $M$ data sources. For the same data request $\mathcal{Q}$, we assume that most nodes and data sources are honest and return a consistent and correct value $d$. Specifically, the adversary can compromise at most $P(D) < \frac{M}{2}$ of the data sources and $P(O) < \frac{N}{2}$ of the oracle nodes, ensuring that honest entities constitute the majority. Meanwhile, the threshold $\mathfrak{t}$ for the threshold signature satisfies $\frac{N}{2} < \mathfrak{t} \leq N$, ensuring that the

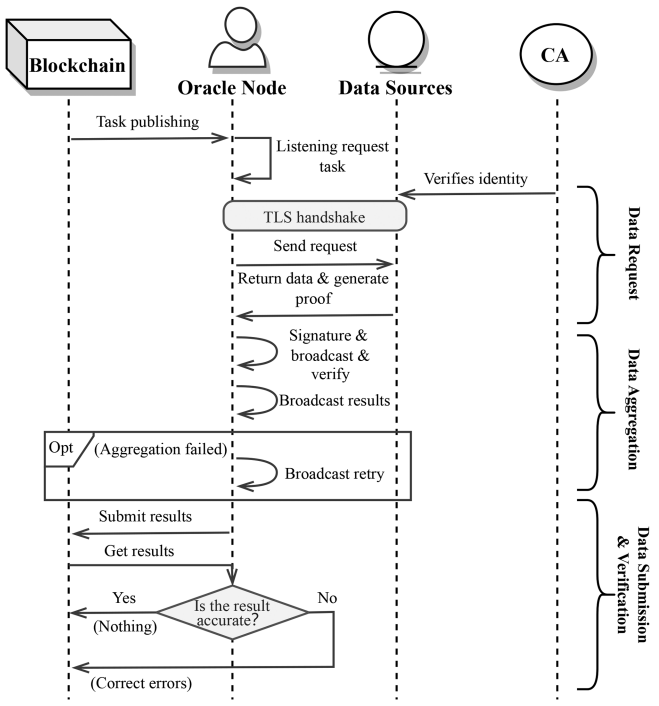



Fig. 3. TBLS protocols.

aggregate signature can only be completed with the participation of an honest majority [29], [30].

## IV. Detail Design

### A. TBLS Protocols

To address challenge 1, we designed a data aggregation protocol TBLS, which ensures data source diversity with low resource overhead, building on the foundation of threshold signature data aggregation methods. The detailed process is shown in Fig. 3.

*a) Data Request Phase:* Node $\mathcal{O}_i$ selects a data source $\mathcal{D}_j$ either randomly or strategically based on the data request task $\mathcal{Q}$ published on the blockchain. The TLS-N protocol is used for network data transmission between $\mathcal{O}_i$ and $\mathcal{D}_j$. It allows $\mathcal{O}_i$ to generate an undeniable proof $\mathcal{P}_i^j$ of the session data source, enabling third parties to verify that the data was indeed obtained by the node from a specific data source. It prevents both parties from repudiating their communication afterward and provides legal traceability [13], [31]. Specifically, $\mathcal{O}_i$ first establishes a TLS connection with the data source and negotiates TLS-N parameters during the handshake process. Then, $\mathcal{O}_i$ utilizes the TLS-N protocol to sign the session content, including handshake data and session keys. After the data transmission is completed, it generates non-repudiable proof $\mathcal{P}_i^j$ for the data source. It signs the obtained data $d_j$ by its private key $sk_i$, resulting in $\sigma_i^j = sign(d_j, sk_i)$.

*b) Data Aggregation Phase:* Node $\mathcal{O}_i$ broadcasts the data and its proof $(\mathcal{I}, \mathcal{P}_i^j, \sigma_i^j)$ and also receives and verifies data and proofs from other nodes. During the aggregation process, the system must verify whether the group signature $\sigma$ from $\mathfrak{t}$ nodes meets the threshold requirements for successful aggregation, as well as whether the associated data source proofs $\{\mathcal{P}_i^j \mid i \in N, j \in M\}$ meet the diversity requirement $K$. The parameter $K$ represents the number of data sources, typically requiring that $\frac{M}{2} < K \leq M$ to ensure data diversity and reliability. Additionally, to reduce resource overhead and communication load in the oracle network, each node is restricted from uploading data for aggregation more than once per task. If multiple uploads are detected, the node will be disqualified from participating in the current round. When the threshold or diversity requirements are unmet or the task times out, a retry request is initiated, and the retry counter is incremented to ensure task uniqueness. During the retry process, nodes must recollect data and verify the diversity of data sources while avoiding the resubmission of previously uploaded data.

TABLE II
Comparison of Two Enhanced Data Validation Mechanisms

| Dimension | Mechanism I | Mechanism II |
|---|---|---|
| Trigger Timing | Immediate after aggregation | Enters confirmation window after aggregation |
| On-chain Load | High (all proofs submitted) | Low (only disputed cases uploaded) |
| Verification Timeliness | High (immediate on-chain check) | Medium (wait $\Delta T$) |
| Resistance to Malicious Submission | Strong (verified before submission) | Strong (dispute-triggered correction) |
| Computation / Communication Cost | High (on-chain cost significant) | Low (mostly off-chain validation) |
| Application Scenario | High-value, security-sensitive tasks | Routine tasks, optimized for performance |

*c) Data Submission and Verification Phase:* Two feasible data validation enhancement mechanisms are adopted to prevent malicious nodes from bypassing the data source diversity constraint when submitting results. This paper primarily adopts Mechanism II. Table II compares the advantages and disadvantages of the two mechanisms.

*Mechanism I:* After successful data aggregation and source diversity verification, $\mathcal{O}_i$ submits its result and group signature $(d, \sigma)$ along with the complete set of data source proofs $\{\mathcal{P}_i^j\}$ to the blockchain, and broadcasts the aggregation result as $(\mathcal{I}, \{\mathcal{P}_i^j\}, \sigma)$. Similar to the DOS Network [27], other nodes stop processing the task once verification succeeds. The oracle contract not only verifies data correctness using the group public key $G_{pk}$ registered on-chain via $verify_{acc}(\sigma, G_{pk}, d)$, but also validates source diversity through $verify_{div}(\{\mathcal{P}_i^j\}, \mathcal{C}, K)$.

*Mechanism II:* $\mathcal{O}_i$ only submits the result and the group signature $(d, \sigma)$ to the blockchain, where the oracle contract performs data verification using $verifyacc(\sigma, G_{pk}, d)$. At this point, the aggregation result enters a confirmation period of duration $\Delta T$ (e.g., 6 blocks) and does not take effect immediately [5], [32]. If $\mathcal{O}_k$ detects that the result has been recorded on-chain but has not received the broadcast $(\mathcal{I}, \{\mathcal{P}_i^j\}, \sigma)$, it initiates a query to the submitting node $\mathcal{O}_i$. If no response is received or the verification fails, it is determined that the task $\mathcal{I}$ lacks sufficient diversity. A dispute broadcast is then immediately triggered in the form $(\mathcal{I}, Tx, \sigma_k^{\mathcal{I}})$, where $Tx$ denotes the transaction address of the aggregation result for task $\mathcal{I}$, and $\sigma_k^{\mathcal{I}} = sign(Tx, sk_k)$ is the dispute signature. Once a threshold group dispute signature $\sigma^{\mathcal{I}}$ is collected, the aggregation result of task $\mathcal{I}$ is revoked, and the misbehaving node is penalized.

### B. Data Source Selection Strategy

Under the constraints of the TBLS aggregation protocol, nodes must carefully select data sources to maximize their chances of successful aggregation and receiving rewards. They must not only consider the response speed of the data source,

aiming to be among the $\mathfrak{t}$ nodes aggregated but also take into account the selection of other nodes. For example, while $\mathcal{O}_1$ may retrieve data from $\mathcal{D}_1$ the fastest, if node $\mathcal{O}_2$ retrieves data even quicker, $\mathcal{O}_1$'s chance of earning a reward diminishes. Moreover, selecting a malicious data source reduces the likelihood of successful aggregation. Thus, nodes must select a data source that is most likely to be successfully aggregated without knowing the selections of other nodes. This decision balances maximizing the node's benefits by ensuring both data source diversity and optimal response speed.

*a) Bayesian Game Model:* Since a node cannot fully grasp the information of other nodes' selection when selecting a data source, we construct the node's data source selection problem as a Bayesian game model to solve challenge 2. The game can be constructed as a quintuple of the form $G = (N, A, \Theta, P, U)$.

- Set of participants $N$ represents the set of oracle nodes participating in the game, i.e. $N = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_N\}$, where $N$ is the total number of nodes and each $\mathcal{O}_i$ represents an individual oracle node.
- Strategy Space $A$ is the set of strategies (data sources) that each participant can select from, i.e. $A = \{a_1, \ldots, a_M\}$. Where $M$ is the total number of data sources, and $a_j$ represents an alternative data source $\mathcal{D}_j$.
- Type set $\Theta$ represents the possible types of each node, with the size of the set corresponding to the total number of data sources, $M$. Formally, $\Theta = \{\theta_1, \theta_2, \ldots, \theta_M\}$, where each type $\theta_i \in \Theta$ indicates a specific advantage a node may have when accessing data source $\mathcal{D}_j$. This advantage could arise from factors such as the node $\mathcal{O}_i$ being geographically closer to $\mathcal{D}_j$, having a faster network connection, or the likelihood of other nodes accessing the same data source is lower, etc.
- Type probability distribution $P$ describes the prior probability distribution of each node's type. Specifically, $P_i(\theta_j)$ represents the probability that node $\mathcal{O}_i$ belongs to type $\theta_j$, based on public prior knowledge and without access to node $\mathcal{O}_i$'s private information (such as response time or historical selections). This probability distribution also referred to as prior belief, reflects the initial estimate of the node's likely type.
- Utility function $U$ represents the benefit or utility of each node given a particular type and strategy selection. For each node $\mathcal{O}_i$ and type $\theta_i$, the utility function $U_i(a_i, \theta_i, \theta_{-i}, a_{-i})$ describes how well the node $\mathcal{O}_i$ is able to select the strategy $a_i$ given its type $\theta_i$, other nodes' types $\theta_{-i}$ and other nodes' strategies $a_{-i}$ when selecting its strategy $a_i$. Here $a_i \in A$ is the strategy of the node $\mathcal{O}_i$, and $a_{-i}$ is the combination of the strategies of other nodes except $\mathcal{O}_i$.

*b) Markov Decision Process:* Due to the dynamic belief updating and incomplete information in the selection game of oracle nodes, the nodes must gradually learn the optimal strategy based on limited observation data, and traditional game theory makes it difficult to deal with such complex interactions [33]. Reinforcement Learning (RL) is suitable for approximating the optimal strategy through continuous trial and error in a dynamic environment, and thus we can convert the problem into a Partially Observable Markov Decision Process (POMDP) [34], [35]. Further, to optimize the solution process, we simplify it to a Markov Decision Process (MDP), denoted as $(S, A, T, R, \gamma)$.

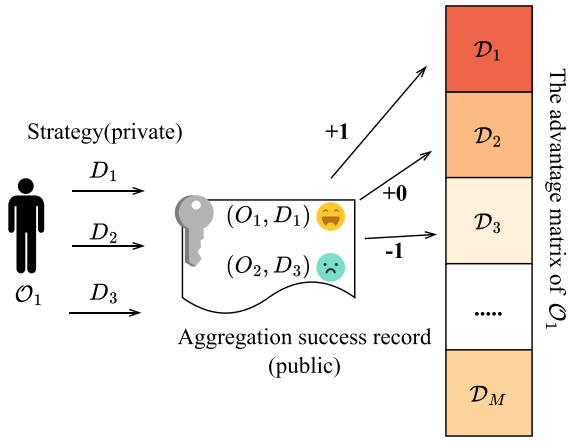



Fig. 4. The influence of different strategies of node $\mathcal{O}_1$ on the advantage matrix.

To transition from partial to full observability, we incorporate type inference, or beliefs, from Bayesian games into the state $S$. Specifically, we design an advantage matrix as state $S$ to represent both observable and unobservable information uniformly. During the construction of the advantage matrix, nodes can continuously update the belief $P_{-i}(\theta)$ according to the strategy they select (unobservable information) and the received aggregation success results (observable information), to improve the accurate inference of other node types.

Each node maintains its advantage matrix $\mathbb{T}$, which tracks its performance across various data sources. By updating this matrix, nodes adjust their belief about the types of other nodes after each task, based on the aggregation results. As illustrated in Fig. 4, when a node successfully aggregates data from a particular source, the corresponding entry in the advantage matrix increases by $+1$, indicating that the node holds an advantage for that data source. Conversely, if the node selects the same data source as others but fails to aggregate, the entry decreases by $-1$, suggesting a lack of advantage. If no other node successfully aggregates data from the select source, no advantage is inferred, and the entry remains 0. This mechanism allows the advantage matrix to reflect the node's private selections and the publicly available aggregation results.

The set of actions $A$ is the same as the strategy space $A$ of the Bayesian game. The state transfer function $T(s'|s, a)$ denotes the probability that the system transfers to the next state $s'$ after selecting action $a$ in the current state $s$. The $\gamma \in (0, 1]$ is a discount factor that represents the current value of the future reward.

The reward $R$ corresponds to the utility function $U$. In this task, $R$ denotes the reward obtained by the node after successful aggregation. The $\mathfrak{t}$ nodes that are successful in aggregation in each task will receive the reward, while the other nodes cannot receive the reward and are agnostic about their data source selection. This reward structure encourages nodes to select the appropriate data source to increase the aggregation success rate

and maximize the reward.

$$R = \begin{cases} 1, & \text{reach consensus,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

*c) Strategy Solving:* In the given Markov Decision Process (MDP) $(S, A, T, R, \gamma)$, we employ the Double Deep Q-Network (DDQN) algorithm [28] to find the optimal action decision policy $\pi^*$. DDQN aims to learn the state-action value function $Q(s, a)$, which satisfies the Bellman equation:

$$Q(s, a) = \mathbb{E}[R_t + \gamma \max_{a'} Q(s_{t+1}, a') | s_t = s, a_t = a]. \tag{2}$$

The goal of DDQN is to approximate the solution to this equation, leading to the optimal policy $\pi^*$.

The detailed steps are as follows: ❶ Initialize the main Q-network $Q_\theta$ and the target Q-network $Q_{\theta^-}$ with parameters $\theta$ and $\theta^-$, respectively. Also, set the experience replay buffer $\mathcal{D}$, exploration rate $\epsilon$, and discount factor $\gamma$, along with other hyperparameters. ❷ At each time step $t$, in the current state $s_t$, select an action $a_t$ using an $\epsilon$-greedy policy:

$$a_t = \begin{cases} \text{random action}, & \text{with probability} \epsilon, \\ \arg\max_a Q_\theta(s_t, a), & \text{with probability } 1 - \epsilon. \end{cases} \tag{3}$$

❸ Execute the selected action $a_t$ in the environment, receiving a reward $r_t$ and observing the next state $s_{t+1}$. ❹ Store the experience tuple $(s_t, a_t, r_t, s_{t+1})$ into the experience replay buffer $\mathcal{D}$. ❺ Randomly sample a batch of experience $(s_i, a_i, r_i, s_{i+1})$ with size $B$ from the buffer $\mathcal{D}$, and calculate the target Q-value $y_i$.

$$y_i = r_i + \gamma Q_{\theta^-}(s_{i+1}, \arg\max_a Q_\theta(s_{i+1}, a)). \tag{4}$$

Here, the target Q-value depends on the estimation of the target network $Q_{\theta^-}$ to mitigate the problem of overestimation. According to the Bellman equation, $y_i$ is an approximation of $Q(s_i, a_i)$.

❻ Update the main Q-network parameters using gradient descent and Mean Squared Error (MSE) loss function, adjusting the parameters $\theta$ of the main Q-network.

$$L(\theta) = \frac{1}{B} \sum_{i=1}^{B} (y_i - Q_\theta(s_i, a_i))^2. \tag{5}$$

❼ Periodically copy the main Q-network parameters $\theta$ to the target network $\theta^-$, ensuring training stability.

By repeating steps ❷ to ❼, DDQN progressively updates the Q-values and eventually converges to an approximation of the state-action value function $Q(s, a)$. The final optimal policy is obtained by selecting the action with the maximum Q-value for each state:

$$\pi^*(s) = \arg\max_a Q(s, a). \tag{6}$$

## V. Experiment and Safety Analysis

We implemented the proposed solution using Python 3.9 to simulate the oracle network. Specifically, we constructed the oracle network comprising 50 oracle nodes and 20 data sources. The oracle nodes were deployed on a platform equipped with an Intel(R) Core(TM) i7-9700F processor and 16 GB of RAM. The parameter settings are presented in Table III. To fully demonstrate the effectiveness of SEMSO, we compared it against the following four baselines:

TABLE III
Experimental Parameter Setting

| Parameter name | Value |
|---|---|
| total oracle nodes $N$ | 50 |
| total data sources $M$ | 20 |
| signature threshold $\mathfrak{t}$ | 20 |
| diversity requirements $K$ | 18 |
| oracle task counts | 1000 |
| network latency distribution (s) | $U(0.1, 2.3)$ |
| learning rate $\eta$ | 5e-4 |
| exploration rate $\epsilon$ | 0.05 |
| discount factor $\gamma$ | 0.99 |
| memory size | 1000 |

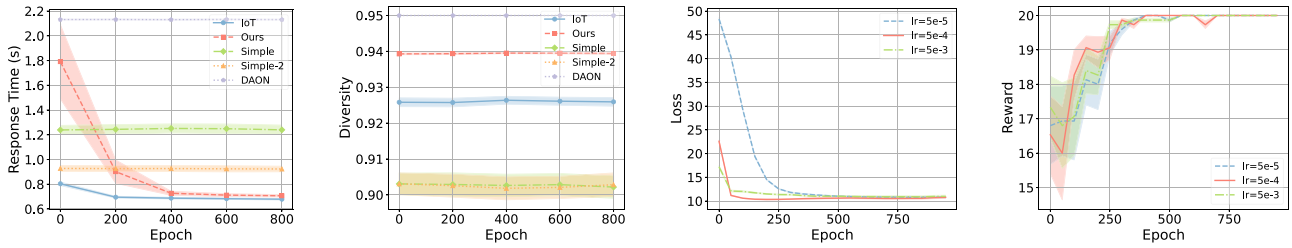
- *DAON [19]:* A representative example of MDS oracle solutions. Similar solutions include Chainlink [9], which ensures data diversity and reliability by requiring each node to traverse and access $M$ data sources.
- *IoT [10]:* The performance improvement scheme of MDS oracle in heterogeneous environments uses the reputation mechanism to allocate $\mathcal{K} = 2$ nodes with fast response and high success rates for different data sources.
- *Simple:* Each node simply selects a data source randomly for access.
- *Simple-n:* Each node simply and randomly selects $n = 2$ data sources to access. (Default Simple-2)

### A. Research Questions (RQs)

To ensure data security and reliability, oracle nodes typically adopt a strategy of redundantly accessing multiple data sources. However, this significantly increases resource overhead and response time. We have designed and conducted a series of experiments to evaluate the proposed method's effectiveness in improving system performance while ensuring data diversity.

These experiments aim to answer the following research questions (RQs):

- *RQ1 - Performance Comparison:* Compared to the baseline solution, can the proposed solution control time overhead and further improve system efficiency while considering the diversity of data sources when nodes obtain data from multiple sources?
- *RQ2 - Principle Analysis:* How does the proposed strategy enable nodes to effectively mitigate the risk of data concentration on a single data source while improving access speed, given that each node accesses a data source only once? This question aims to explore the fundamental reasons behind the effectiveness of the proposed method.
- *RQ3 - Usability Analysis:* Does the proposed method introduce potential risks in terms of security and scalability, which could jeopardize the overall availability of the system? This question aims to comprehensively evaluate the robustness performance of the method in practical applications.




(a) Response Time (b) Diversity of Data Sources (c) Loss (d) Reward

Fig. 5. Baseline comparison, diversity and response time, effectiveness of proposed reinforcement learning algorithms.

### B. RQ1 - Performance Comparison

We measure the diversity of data sources using a variant of the information entropy formula, $D = 1 - \sum_{i=1}^{n} p_i^2$. where $p_i$ is the frequency of occurrence of the category $i$. The value of $D$ ranges from $D \in [0, 1 - \frac{1}{n}]$, $n$ is the total number of categories in the final aggregated data. When the frequencies of all categories are uniformly distributed, $D$ will approach the maximum value $1 - \frac{1}{n}$. In the same case, the value of $D$ is 0.

Fig. 5(a) and (b) illustrate the node response time and data source diversity under the experimental setup for different approaches. Due to the redundant access to multiple data sources, DAON is adversely impacted by the cask effect. Each task is required to wait for the slowest access result to be returned, resulting in an overall response time that consistently lags behind that of alternative solutions. It is evident that in the Simple and IoT schemes, nodes do not need to aggregate data from all sources, which leads to a significant advantage in response time. However, this comes at the cost of reduced data source diversity. In particular, the IoT approach achieves excellent performance by using reputation contracts to assign nodes with fast response times for each data source. However, its major drawback is the reliance on a TEE to ensure the credibility of the node response times, as well as the substantial storage and computation overhead on the blockchain to calculate node reputations. In contrast, SEMSO achieves comparable response times to IoT without the need for TEE support or increased on-chain computation and storage costs, while also ensuring higher data source diversity. This demonstrates that the proposed scheme effectively addresses Challenges 1 and 2, enhancing system efficiency while maintaining the diversity of data sources.

Fig. 5(c) and (d) shows the proposed reinforcement learning method's loss and reward curves under different learning rates. In the case of $M = 20, \mathrm{t} = 20$, the rewards under different settings eventually approach 20, i.e., there are data from at least $K$ different data sources in each aggregation such that t nodes are rewarded. The proposed method demonstrates strong usability and can eventually learn and approximate the optimal policy, depending on the environment, even when different learning rates are set.

Table IV presents the average time consumption over the last 100 tasks for different schemes. Although the proposed solution introduces RL to learn the optimal data source selection strategy, the design of the advantage matrix simplifies the RL training process. Nodes only require minimal computational resources to significantly reduce communication time. As a result, SEMSO achieves response times comparable to the IoT approach without incurring significant on-chain computation and storage costs or relying on the TEE. Compared to other approaches like DAON and Simple, which also do not require TEE, SEMSO reduces response time by 23.5% relative to the best baseline (Simple-2).

TABLE IV
OVERALL TIME CONSUMPTION ANALYSIS OF ORACLE NODES

| | Ours | Simple-1 | Simple-2 | IoT | DAON |
|---|---|---|---|---|---|
| Computational time(s) | 0.0025(±0.0003) | - | - | - | - |
| Correspondence(s) | 0.692(±0.037) | 1.219(±0.210) | 0.908(±0.126) | 0.672(±0.036) | 2.111(±0.007) |
| Total(s) | **0.695(±0.037)** | 1.219(±0.210) | 0.908(±0.126) | **0.672(±0.036)** | 2.111(±0.007) |

TABLE V
NUMBER OF DATA SOURCE ACCESSES AND ADDITIONAL OVERHEAD FOR DIFFERENT SCHEMES

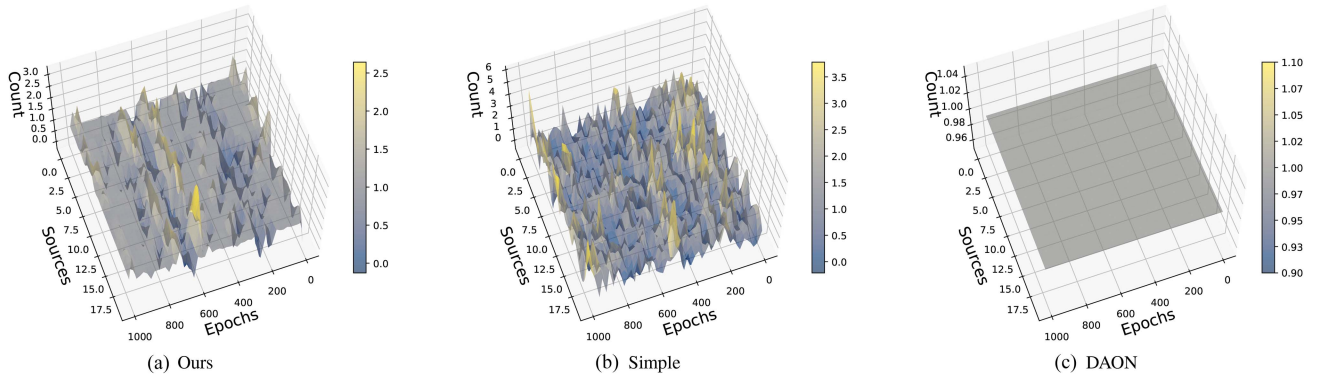
| | Ours | Simple-1 | Simple-n | IoT | DAON |
|---|---|---|---|---|---|
| Number of Data Source Accesses | 1 | 1 | $n$ | $(0 - M \times K)$ | $M$ |
| Additional On-chain Overhead | ✘ | ✘ | ✘ | Node Selection & State Store | ✘ |
| Additional Technical Support | ✘ | ✘ | ✘ | TEE | ✘ |

Table V shows the number of data source accesses and the additional overhead for different schemes. Compared to traditional MDS oracle solutions such as DAON and IoT, SEMSO requires each node to only access a data source. It is particularly important for data sources that charge fees based on the number of accesses [36]. Furthermore, in contrast to performance-optimized solutions like IoT, SEMSO does not introduce additional on-chain overhead or require specialized technical support, significantly reducing gas consumption and the cost of acquiring TEE devices, thereby greatly improving usability.

**Answer for RQ1:** Compared to the baseline plan, the proposed solution not only performs well in terms of data source diversity but also has significant advantages in performance improvement.

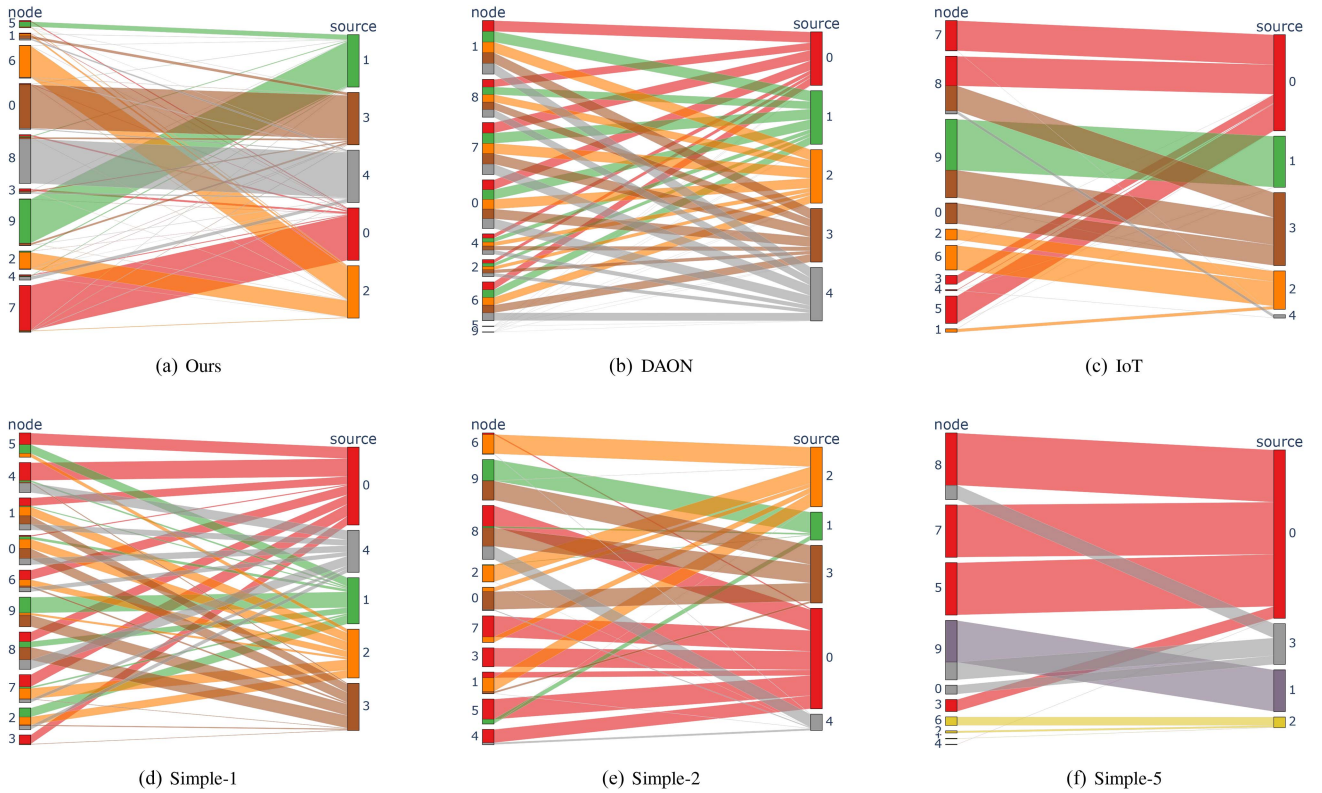
### C. RQ2 - Principle Analysis

*1) Diversity Analysis:* Fig. 6 shows the distribution of the final aggregated data sources over 1000 tasks. First of all, DAON, as a representative of the traditional MDS scheme, guarantees that the final aggregated results are completely from different data sources by redundantly accessing all the data. On the other hand, SEMSO can set $K = 18$ in TBLS to constrain the source diversity of the final aggregated data. Compared with




(a) Ours (b) Simple (c) DAON

Fig. 6. The significance of diversity in the number of data sources selected for each task.




(a) Ours (b) DAON (c) IoT

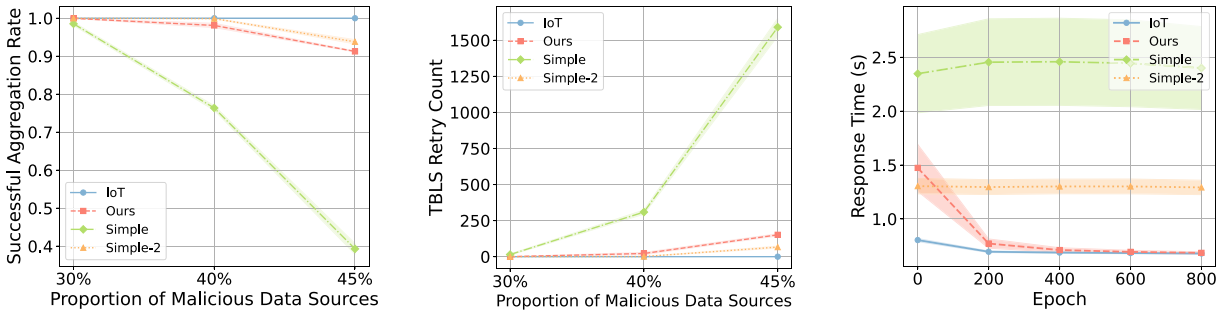
(d) Simple-1 (e) Simple-2 (f) Simple-5

Fig. 7. Access of nodes to data sources in different scenarios. For example, 10 nodes access 5 data sources 1000 times.

completely randomly selecting data sources without constraints, the diversity of data sources is higher and the final data sources are more balanced. It verifies the effectiveness of TBLS in solving Challenge 1. As the epoch increases, the data source becomes more and more even, inhibiting the possibility of data being manipulated, which further confirms the effectiveness of SEMSO.

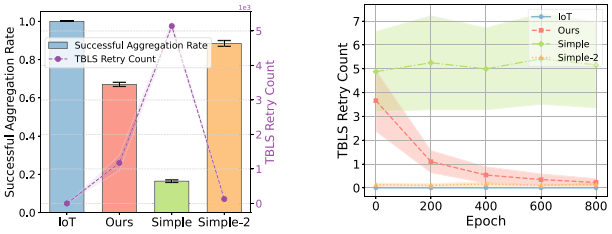
In Fig. 7, we further analyze the data source access patterns of different schemes. Using $N = 10$, $M = 5$, and $\mathfrak{t} = 5$ as an example, where 10 oracle nodes perform 1000 tasks from 5 data sources, Simple-1, Simple-2, and Simple-5 represent cases where nodes randomly select $\{1, 2, 5\}$ data sources for access, respectively. Simple-5 represents an extreme case focusing on minimizing response time, where each node accesses data from all 5 data sources and broadcasts the results for aggregation. The fastest $\mathfrak{t} = 5$ responses are selected for final aggregation.

As shown in Fig. 7, the final aggregated data sources for schemes like IoT and Simple, which do not consider data source diversity, exhibit an uneven distribution. In particular, the Simple-5 scheme, which entirely focuses on achieving rapid responses, may produce erroneous aggregated results if data sources 0 and 3 conspire to return incorrect results, even with the use of the BLS protocol. In contrast to DAON, our proposed scheme does not require fetching data from all $M$ sources, significantly reducing resource overhead. Moreover, similar to IoT, our scheme identifies implicit matching relationships between node 9 and data source 1 (indicated by the green band), which greatly enhances its response speed. However, to ensure data




(a) TBLS Successful Aggregation Rate (b) TBLS Retry Count (c) Response Time

Fig. 8. Aggregation success rate, number of retries, response time when both use TBLS.




(a) TBLS Successful Aggregation Rate & Retry Count (b) TBLS Retry Count

Fig. 9. Aggregation success rate, number of retries in extreme cases with TBLS.

source diversity, our approach avoids situations where a large number of nodes access data from source 0, thus mitigating the risk of aggregation errors mentioned above. The results indicate that SEMSO not only ensures data source diversity without concentrating on specific sources but also enhances response speed by selecting sources that can provide quick responses, similar to IoT.

*2) Effectiveness of the Proposed Data Source Selection Strategy:* To analyze the effectiveness of the proposed data source selection strategy, we replace the BLS protocol used by IoT, Simple, etc. schemes with TBLS for comparison. DAON needs to perform aggregation operations at nodes and cannot adapt to TBLS.

Fig. 8(a) and (b) illustrates the aggregation success rates and the number of retries for TBLS tasks under the condition that all schemes use the TBLS aggregation protocol and $K = 11$. When the final number of aggregated data sources falls below $K$, the TBLS protocol will fail and initiate a retry. As the proportion of malicious data sources increases, the aggregation success rate declines while the number of retries escalates. Although the IoT approach, which allocates nodes to data sources, does not experience aggregation failures, the on-chain allocation of nodes incurs significant computational and storage overhead. In contrast, our proposed data source selection method shows a clear advantage over Simple-1 approaches and is comparable to Simple-2. Similarly, Fig. 8(c) demonstrates the notable advantage of our proposed scheme in terms of response time.

Fig. 9 illustrates the aggregation success rates and the number of retries for various schemes using TBLS under extreme conditions where $K = M = 20$. First, both the aggregation success rate and the number of retries, as shown in Fig. 9(a), demonstrate results similar to those observed when $K = 11$. Fig. 9(b) provides a detailed overview of how the number of retries changes over time. As the number of epochs increases, our data source selection strategy gradually learns to select suitable data sources for nodes, thereby avoiding aggregation failures. It indicates that our proposed method not only focuses on improving response times but also enhances the aggregation success rate of TBLS, even in scenarios where complete information about other nodes is not available.

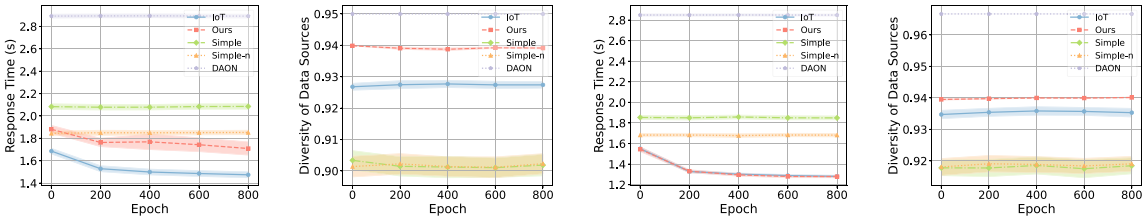
By analyzing Figs. 8 and 9, we observe that as $K$ increases from 11 to 20, the number of TBLS aggregation failures also rises. A higher $K$ enhances data source diversity, thereby improving the system's resistance to manipulation and overall security. However, it also reduces the aggregation success rate and increases the number of retries due to failed attempts. To maintain system liveness, $K$ should be carefully configured based on the specific requirements of the application, balancing security and operational feasibility.

**Answer for RQ2:** The proposed TBLS method enhances the diversity of final data sources by introducing data source verification, thereby reducing the risk of data manipulation. Additionally, the proposed data source selection strategy improves data aggregation success rates and system performance by learning and uncovering potential matching relationships between nodes and data sources.

### D. RQ3 - Usability Analysis

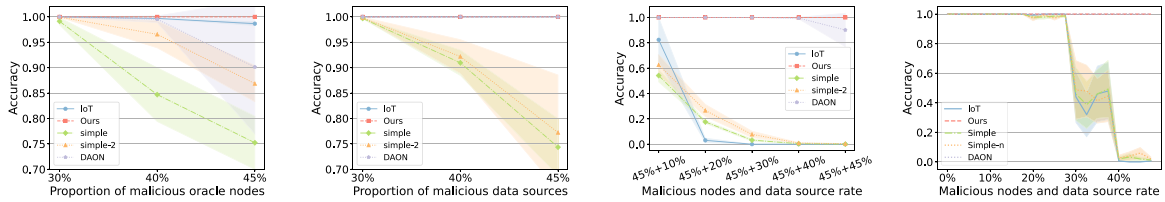
*1) Robustness:* To verify the robustness of the proposed scheme, we analyze the effectiveness of the proposed scheme under different network delay distributions and different network sizes.

Fig. 9(a) and (b) illustrate the response times and data source diversity of different schemes under network delays following a Gaussian distribution $N(2.0, 0.4)$, with results resembling those seen in the previous random distribution scenarios. Additionally, Fig. 9(c) and (d) demonstrate the performance when the network scale is expanded to 100 nodes and the number of data sources $M$ increases to 30 while maintaining the thresholds $t = 20$ and $K = 18$. In this environment, SEMSO's response time becomes




(a) Response Times (b) Diversity of Data Sources (c) Response Times (d) Diversity of Data Sources

Fig. 10. Effectiveness in different network environments and network sizes.




(a) Malicious Nodes (b) Malicious Data Sources (c) Malicious Nodes & Data Sources (d) Dynamic malicious proportion

Fig. 11. The accuracy of data under varying degrees of malicious nodes and data sources.

closer to that of IoT, while still maintaining greater diversity than IoT. It indicates the robustness of SEMSO under varying conditions.

*2) Attack Defence:* Fig. 11 demonstrates the ability of different schemes to defend against the malicious behavior of nodes and data sources when they conspire to return incorrect results. For example, if the correct answer is '5', malicious nodes and data sources may collude to jointly return an incorrect answer, such as '10'. As Fig. 11(a), when the proportion of malicious nodes increases, the data accuracy of other baselines gradually decreases, especially when the number of malicious nodes $0.45 \times N \approx 23$ is greater than the threshold $\mathfrak{t} = 20$, the IoT and DAON methods cannot guarantee the accuracy of their data. The proposed scheme, on the other hand, can ensure the integrity and accuracy of the data uploaded by the nodes with the help of the proofs generated by the TLS interaction process.

Fig. 11(b) demonstrates the impact of the proportion of malicious data sources on the accuracy of the final data, as the problem of data aggregation error exposed in Fig. 7 when the proportion of malicious data sources increases, the schemes such as Simple are not able to guarantee the accuracy of the data they acquire. In contrast, SEMSO and schemes such as DAON guarantee the accuracy of their data due to the diversity of their data sources.

Fig. 11(c) shows the data accuracy of different schemes in the extreme case when the nodes conspire with the data source to return the incorrect data. As the proportion of malicious nodes increases, the accuracy rate of schemes such as IoT and Simple decreases rapidly. Even DAON returns incorrect data because the number of malicious nodes is larger than the threshold $\mathfrak{t}$. In contrast, under the fundamental assumption that the number of malicious nodes is less than $\frac{N}{2}$ and the number of malicious data sources is less than $\frac{M}{2}$, the proposed scheme is still able to maintain data correctness.

Fig. 11(d) illustrates the impact of the decreasing ratio of honest to malicious devices in real-world networks. Every 200 task rounds, we increase the proportion of malicious nodes and data sources by 10%. It can be observed that as the proportion of malicious devices increases, schemes such as IoT struggle to ensure the correctness of the final aggregated data. In contrast, our proposed scheme guarantees data reliability through dual verification of both the data and its source, even in scenarios where both nodes and data sources are unreliable.

*3) Security Analysis:* Based on the security assumptions proposed in Section III, we briefly analyze how the system ensures its security. Existing studies have already proven the security of BLS and TLS-N [29], [37]. Therefore, our focus will be on analyzing how the TBLS mechanism ensures data correctness and source diversity.

*a) Theorem (Data Correctness):* If the number of malicious data sources is less than $\frac{M}{2}$ and $K > \frac{M}{2}$ as set by TBLS, SEMSO will guarantee the accuracy of the final aggregated data.

*Proof.* First, the nodes exchange data with the data source through the TLS-N protocol, and the proofs generated in this process ensure the integrity of the data and prevent the nodes from tampering with the data. Therefore, even if all nodes $j$ receive data $d'$ from malicious data sources $\mathcal{M}_{\text{malicious}}$, the differences between the data $d'$ provided by these malicious data sources and that provided by the honest data sources lead to the fact that even if $\mathfrak{t}$ fragments of the same data signatures are collected during the aggregation process $\sigma_{j,d'}$, the diversity requirement $K$ cannot be satisfied due to $\mathcal{M}_{\text{malicious}} < \frac{M}{2} \leq K$, thus preventing successful aggregation. TBLS effectively prevents the influence of malicious data sources on the data aggregation process.

*b) Theorem (Data Source Diversity):* If the number of data sources contributing to the final aggregated result is less than the predefined diversity threshold $K$, the TBLS scheme will fail to aggregate the signature. This ensures that the final submitted data is aggregated from a sufficiently diverse set of data sources.

*Proof.* First, all data sources provide data to the nodes using the TLS-N protocol. The proofs generated by TLS-N guarantee data authenticity and verifiable provenance, ensuring that each piece of data can be traced back to its specific source while preventing intermediary nodes from forging data origins. During the signature aggregation process, TBLS imposes an additional data source diversity constraint on top of the threshold signature constraint $t$. Specifically, at least $t$ nodes must sign the same data, and the number of distinct data sources contributing to the signed data must be no less than $K$. Only if both conditions are met will the aggregation succeed and be considered valid. Even if the threshold signature verification passes, the submitted content will not be accepted if the diversity verification fails.

**Answer for RQ3:** Experiments and security analysis show that the proposed scheme maintains excellent system performance while ensuring data diversity, especially in terms of scalability and security.

## VI. Conclusion

This paper presents a secure and efficient multi-data source oracle solution, SEMSO, which reduces resource consumption and response times while ensuring data source diversity. To build a low-cost, distributed trust system between nodes and data sources, we design a novel off-chain data aggregation protocol TBLS. Then, under the assumption of rational agents, we model the process of node data source selection as a Bayesian game and apply reinforcement learning to solve it. This approach maximizes node utility while minimizing system resource consumption and response times. Both experimental results and security analysis validate the reliability and effectiveness of the proposed solution.

## References


[1] F. Schär, "Decentralized finance: On blockchain-and smart contract-based financial markets," *FRB St Louis Review*, vol. 103, no.nbsp;2, pp. 153–174, 2021.

[2] J. Zhu, T. Feng, Y. Lu, and W. Jiang, "Using blockchain or not? a focal firm's blockchain strategy in the context of carbon emission reduction technology innovation," *Bus. Strategy Environ.*, vol. 33, no. 4, pp. 3505–3531, 2024.

[3] J. Andrew, D. P. Isravel, K. M. Sagayam, B. Bhushan, Y. Sei, and J. Eunice, "Blockchain for healthcare systems: Architecture, security challenges, trends and future directions," *J. Netw. Comput. Appl.*, vol. 215, 2023, Art. no. 103633.

[4] A. Hassan, I. Makhdoom, W. Iqbal, A. Ahmad, and A. Raza, "From trust to truth: Advancements in mitigating the blockchain oracle problem," *J. Netw. Comput. Appl.*, vol. 217, 2023, Art. no. 103672.

[5] J. Peterson, J. Krug, M. Zoltu, A. K. Williams, and S. Alexander, "Augur: A decentralized oracle and prediction market platform (v2. 0)," 2019. [Online]. Available: https://augur.net/whitepaper.pdf

[6] R. Berryhill and A. Veneris, "Astraea: A decentralized blockchain oracle," *Proc. IEEE Int. Conf. Int. Things IEEE Green Comput. Commun. IEEE Cyber, Phys. Social Comput.*, pp. 1145–1152, 2018.

[7] P. Liu, Y. Xian, C. Yao, P. Wang, L.-E. Wang, and X. Li, "A trustworthy and consistent blockchain oracle scheme for industrial Internet of Things," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 5, pp. 5135–5148, Oct. 2024.

[8] Y. Lin et al., "A novel architecture combining oracle with decentralized learning for IIoT," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 3774–3785, Mar. 2023.

[9] L. Breidenbach et al., "Chainlink 2.0: Next steps in the evolution of decentralized oracle networks," *Chainlink Labs*, vol. 1, pp. 1–136, 2021.

[10] Y. Xian, L. Zhou, J. Jiang, B. Wang, H. Huo, and P. Liu, "A distributed efficient blockchain oracle scheme for Internet of Things," *IEICE Trans. Commun.*, vol. E107-B, no. 9, pp. 573–582, 2024.

[11] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 270–282.

[12] C. Liu et al., "Extending on-chain trust to off-chain–trustworthy blockchain data collection using trusted execution environment (TEE)," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3268–3280, Dec. 2022.

[13] H. Ritzdorf, K. Wust, A. Gervais, G. Felley, and S. Capkun, "TLS-N: Non-repudiation over TLS enabling ubiquitous content signing," in *25th Annual Network and Distributed System Security Symposium*, San Diego, California, USA: The Internet Society, 2018. doi: 10.14722/ndss.2018.23272.

[14] M. Iqbal, A. Chiarelli, and R. Matulevičius, "Bridging two worlds: Framework for secure implementation of blockchain oracles," in *Proc. 2024 IEEE Int. Conf. Softw. Anal., Evol. Reengineering-Companion*, 2024, pp. 12–22.

[15] C. Team, "Oracle manipulation attacks are rising, creating a unique concern for DEFI," 2023. [Online]. Available: https://www.chainalysis.com/blog/oracle-manipulation-attacks-rising/

[16] P. Lv, X. Zhang, J. Liu, T. Wei, and J. Xu, "Blockchain oracle-based privacy preservation and reliable identification for vehicles," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, 2021, pp. 512–520.

[17] K. Almi'Ani, Y. C. Lee, T. Alrawashdeh, and A. Pasdar, "Graph-based profiling of blockchain oracles," *IEEE Access*, vol. 11, pp. 24995–25007, 2023.

[18] L. Gigli, I. Zyrianoff, F. Montori, C. Aguzzi, L. Roffia, and M. Di Felice, "A decentralized oracle architecture for a blockchain-based IoT global market," *IEEE Commun. Mag.*, vol. 61, no. 8, pp. 86–92, Aug. 2023.

[19] J. Dong, C. Song, Y. Sun, and T. Zhang, "DAON: A decentralized autonomous oracle network to provide secure data for smart contracts," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 5920–5935, 2023.

[20] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A decentralized truth discovery approach to the blockchain oracle problem," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.

[21] A. Pasdar, Y. C. Lee, and Z. Dong, "Connect API with blockchain: A survey on blockchain oracle implementation," *ACM Comput. Surv.*, vol. 55, no. 10, pp. 1–39, 2023.

[22] M. D. Gennaro, L. Italiano, G. Meroni, and G. Quattrocchi, "Deepthought: A reputation and voting-based blockchain oracle," in *Proc. Serv.-Oriented Comput.: 20th Int. Conf.*, Seville, Spain, 2022, pp. 369–383.

[23] S. Woo, J. Song, and S. Park, "A distributed oracle using intel SGX for blockchain-based IoT applications," *Sensors*, vol. 20, no. 9, 2020, Art. no. 2725.

[24] Z. Wang, S. Yiu, and L. Lan, "Multi-signature and game based blockchain interoperability oracle," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 4, pp. 3930–3943, 2025, doi: 10.1109/TDSC.2025.3542079.

[25] F. Zhang, D. Maram, H. Malvai, S. Goldfeder, and A. Juels, "DECO: Liberating web data using decentralized oracles for TLS," in *Proc. 2020 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 1919–1938.

[26] Z. Luo, Y. Jia, Y. Shen, and A. Kate, "Proxying is enough: Security of proxying in TLS oracles and aead context unforgeability," *Cryptol. ePrint Arch.*, 2024, Art. no. 733. [Online]. Available: https://eprint.iacr.org/2024/733

[27] D. Network, "A decentralized oracle service network to boost blockchain usability with real world data and computation power," 2019. [Online]. Available: https://s3.amazonaws.com/whitepaper.dos/DOSNetworkTechnicalWhitepaper.pdf

[28] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, Art. no. 2094-2100.

[29] R. Bacho and J. Loss, "On the adaptive security of the threshold BLS signature scheme," in *Proc. 2022 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2022, pp. 193–207.

[30] Y. Xie et al., "Accountable and secure threshold EeDSA signature and its applications," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 7033–7046, 2024.

[31] E. U. C. Signing, "TLS-N: Non-repudiation over TLS enabling ubiquitous content signing," *25th Annu. Net. Dist. Syst. Secur. Symp.*, San Diego, California, USA, 2018, doi: 10.14722/ndss.2018.23272.
[32] Tellor, "Tellor," 2021. [Online]. Available: https://docs.tellor.io/tellor/whitepaper/
[33] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*, vol. 2. MIT Press, 1998.
[34] J. Liang, M. Ma, and X. Tan, "GaDQN-IDS: A novel self-adaptive IDS for VANETs based on Bayesian game theory and deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12724–12737, Aug. 2022.
[35] P. G. Morato, C. P. Andriotis, K. G. Papakonstantinou, and P. Rigo, "Inference and dynamic decision-making for deteriorating systems with probabilistic dependencies through Bayesian networks and deep reinforcement learning," *Rel. Eng. Syst. Saf.*, vol. 235, 2023, Art. no. 109144.
[36] S. N. S. Ellis and A. Juels, "Chainlink a decentralized oracle network," 2017. [Online]. Available: https://chain.link
[37] H. Krawczyk, K. G. Paterson, and H. Wee, "On the security of the TLS protocol: A systematic analysis," in *Proc. Annu. Cryptol. Conf.*, 2013, pp. 429–448.

**Youquan Xian** received the master's degree from Guangxi Normal University, in 2024. He has published multiple papers in journals and conferences such as *IEEE Transactions on Network and Service Management*, *Future Generation Computer Systems*, *IEICE Transactions on Communications*, ASPLOS 2025, IEEE SMC 2024, and WASA 2024. His main research includes blockchain and federated learning.

**Xueying Zeng** received the bachelor's degree from Guangxi Science and Technology Normal University, in 2022. She is currently working toward the master's degree from Guangxi Normal University. Her research interests are blockchain, crowdsourcing.

**Chunpei Li** received the PhD degree from the School of Computer Science and Engineering, Guangxi Normal University, in 2024. He is currently conducting postdoctoral research with the Ministry of Education Key Laboratory of Educational Blockchain and Intelligent Technology at Guangxi Normal University. His research interests include blockchain, artificial intelligence, and information security.

**Peng Wang** received the master's degree from the Guilin University of Technology, in 2018. He is currently working toward the doctor's degree with Guangxi Normal University. His research interests include blockchain, data fusion, and data security.

**Dongcheng Li** received the master's degree in sofrware engineering from Guangxi normal university. He is currently working toward the department of Computer Science and Engineering of Guangxi normal university, China. His main research interests include blockchain, data security and recommendation system.

**Peng Liu** received the PhD degree from Beihang University, China, in 2017. He began his academic career as an assistant professor with Guangxi Normal University, in 2007 and was promoted to full professor, in 2022. His current research interests are focused on federated learning and blockchain.

**Xianxian Li** received the PhD degree from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2002. He worked as a professor with Beihang University during 2003-2010. He is currently a professor with the School of Computer Science and Engineering, Guangxi Normal University, Guilin, China. His research interest includes information security.