frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 02 October 2025
DOI 10.3389/fcomp.2025.1670238



OPEN ACCESS

EDITED BY
Zhou Zhou,
Changsha University, China

REVIEWED BY
Jianhua Xiao,
Huaihua University, China
Xianfang Tang,
School of Mechanical Engineering and Automation, China

*CORRESPONDENCE
Yu Liang
leungyu@szpu.edu.cn

RECEIVED 21 July 2025
ACCEPTED 17 September 2025
PUBLISHED 02 October 2025

CITATION
Wang P, Wang Y, Zhang Y, Lan Y, Huang Z, Tang D and Liang Y (2025) Market malicious bidding user detection based on multi-agent reinforcement learning.
*Front. Comput. Sci.* 7:1670238.
doi: 10.3389/fcomp.2025.1670238



# Market malicious bidding user detection based on multi-agent reinforcement learning

Peng Wang[1], Yimeng Wang[1], Yilin Zhang[1], Yin Lan[1], Ziyang Huang[1], Di Tang[1] and Yu Liang[2]*

[1]School of Electronic Information, Hunan First Normal University, Changsha, Hunan, China, [2]School of Artificial Intelligence, Shenzhen Polytechnic University, Shenzhen, Guangdong, China

With the rapid growth of e-commerce and online auction markets, malicious bidding activities have severely disrupted market order. Traditional detection methods face limitations due to their inability to effectively address the covert nature, dynamic characteristics, and massive data volumes associated with such behaviors. To address this challenge, this paper proposes a detection method for users engaging in malicious bidding based on Multi-Agent Reinforcement Learning (MARL). This approach first models target users as specialized agents, then integrates their historical bidding data, and finally learns optimal strategies through competitive games with adversarial agents. Additionally, this paper designs a dynamic adjustment mechanism for the maliciousness coefficient to simulate user behavior changes, enabling precise assessment of malicious intent. Compared to existing fraud detection approaches based on reinforcement learning, the fundamental innovation lies not merely in applying MARL technology, but in introducing the novel "dynamic maliciousness coefficient" mechanism. This mechanism achieves dynamic and precise maliciousness assessment through mathematical modeling and real-time iteration, addressing the shortcomings of traditional MARL models in capturing user behavioral heterogeneity. Experimental results demonstrate that this method exhibits higher detection accuracy and adaptability in complex dynamic market environments. It effectively captures bidder interaction relationships, significantly enhancing the detection of malicious behavior.

KEYWORDS

**bidding detection, dynamic maliciousness coefficient, malicious bidding, market order, multi-agent reinforcement learning**

## 1 Introduction

With the rapid development of global e-commerce and online auction markets, malicious bidding activities have become a significant threat to market integrity and fair competition (Guo et al., 2021). The widespread adoption of online auction platforms and advertising bidding systems, especially the evolution of real-time bidding (RTB) systems which have notably enhanced bidding efficiency, has unfortunately also provided an ideal breeding ground for malicious activities (Cao, 2022). Such malicious behaviors directly lead to bidding outcomes diverging from actual supply-demand dynamics, resulting in price distortions (Gupta and Gupta, 2023). Currently, the traditional methods for detecting malicious bidding in the market mainly consist of rule-based statistical models and supervised learning-based classification models. Conventional rule-based detection methods identify abnormal data that deviates from normal bidding behavior through predefined thresholds or rules. However, in the face of complex and dynamic market

environments, the limitations of this approach are becoming increasingly evident (Mo et al., 2024). In contrast, multi-agent systems can capture the complex interactions among bidders through agent-to-agent game learning and dynamically adjust detection strategies (Kanmaz and Surer, 2020). Each agent's behavior in the market environment is influenced not only by its own strategy but also by the strategies of other bidders. Therefore, by simulating agent interactions, the system can more effectively replicate the competitive dynamics of real markets, thus enhancing detection accuracy (Chen et al., 2024).

To better understand why traditional methods struggle and multi-agent systems hold advantages, it is essential to first clarify what constitutes malicious bidding—including its core definition, key characteristics, and specific forms—since a precise grasp of the target behavior lays the groundwork for evaluating and optimizing detection approaches. Malicious bidding refers to actions taken by certain market participants who use improper means to influence competitive outcomes. It is characterized by high concealment and complexity, making it difficult to effectively detect through traditional monitoring methods. Its common manifestations can be classified into four types: 1. Fake Bidding: Market bids without genuine purchase intent artificially inflate prices, distorting market pricing and forcing other bidders to bear inflated costs; 2. Bid-rigging (collusive bidding): Involving multiple participants who collude in advance or through secret agreements to take turns bidding, ensuring that a specific party wins at the lowest price. This violates the principles of fair competition and disrupts the rational allocation of resources; 3. Fake demand: Mass-generating false bids to create an illusion of high demand, inducing other bidders to pay inflated prices. This is particularly common in advertising bidding systems and commodity auction platforms (Chen et al., 2024); 4. Malicious disruption: Interfering with other bidders' normal transactions through tactics such as malicious bidding or negative reviews, undermining the foundation of fair market competition.

A key prerequisite for ensuring the effectiveness of the aforementioned detection framework–especially in accurately identifying malicious behaviors and avoiding misclassifying legitimate activities–is clarifying the boundary between malicious bidding and normal strategic bidding, as the two may exhibit overlapping behavioral traits that could lead to detection ambiguity without clear differentiation criteria. It is crucial to distinguish malicious bidding from normal strategic bidding in bidding and online auction markets. The fundamental difference between the two lies in "behavioral legitimacy" and "purpose legitimacy", which requires precise differentiation based on two core dimensions: behavioral purpose and behavioral characteristics. 1. Behavioral Purpose: Malicious bidding aims to manipulate the market through improper means and obtain illegal gains, ignoring the principles of fair competition and value matching. In contrast, normal strategic bidding involves enterprises adhering to market rules and commercial logic to achieve supply-demand matching through compliant competition. For example, manufacturers participate in raw material bidding to control costs, or government departments organize auctions for efficient public resource allocation; 2. Behavioral Characteristics: Malicious bidding shows multi-dimensional anomalies, such as bids significantly deviating from the market value or an abnormally high bidding frequency within specific time periods. Normal strategic bidding, on the other hand, conforms to commercial logic, with bids fluctuating around "cost plus reasonable profit" and having clear cost justifications.

Even with a clear understanding of the differences between malicious and normal strategic bidding, the practical implementation of detection remains constrained by inherent flaws in existing technologies—flaws that hinder their ability to fully leverage such differentiation criteria and address real-world detection challenges. Although existing technologies can identify abnormal behavior to some extent, there are four key limitations restricting their practical application: 1. Data Annotation Challenges: Supervised learning depends on large amounts of labeled data. However, malicious bidding shows diverse forms, making comprehensive annotation difficult and limiting the generalization ability of the model; 2. Model Training Complexity: As market data expands, especially in RTB systems, the complexity of data processing and analysis grows exponentially. Traditional machine learning methods struggle to meet the real-time application requirements; 3. Poor Adaptability to Dynamic Environments: Traditional models based on rules and statistics have difficulty coping with evolving market conditions. Malicious bidders adjust their strategies to avoid detection, and static models cannot quickly adapt to dynamic changes; 4. Challenges in Addressing Cross: Platform Malicious Behavior: Malicious bidding may occur across platforms. Differences in data formats between platforms, as well as technical and legal barriers to data sharing, increase the difficulty of detection.

However, the high concealment and diverse manifestations of malicious bidding, as outlined above, pose significant challenges to existing detection practices. Especially traditional methods that already face inherent drawbacks in addressing such complex behaviors. In the field of malicious bidder detection, traditional methods like manual review are costly and inefficient. Rule-based approaches lack flexibility, have difficulty adapting to complex and dynamic market environments, and have poor real-time performance, failing to respond promptly to malicious bidding activities. To overcome these limitations, this paper conducts in-depth research, with contributions in two main aspects: methodological innovations and experimental outcomes. 1. Method Contribution: We propose a detection framework for malicious bidding users based on MARL. This framework models the historical behavior of the users under investigation, treating them as a distinct agent within a multi-agent system. By constructing a game-theoretic interaction process between this agent and other agents in the system, we enable the learning and optimization of user bidding strategies, thus providing a core technical approach for identifying malicious behavior. Additionally, we design a dynamic adjustment mechanism for the maliciousness coefficient. To address the accuracy issue in evaluating malicious bidder behavior, this mechanism dynamically updates the corresponding maliciousness coefficient by analyzing behavioral trends across multiple bidding rounds. At the same time, it simulates the potential "repentance" behaviors of malicious bidders, making the logical design of the detection model more in line with the complexity of user behavior in real-world scenarios. The "repentance" behavior of malicious bidders fundamentally reflects strategic adaptability. This research

indicates that bidders dynamically adjust their bidding strategies based on market feedback and historical outcomes, including shifting from aggressive bidding (similar to malicious price inflation) to conservative bidding (resembling "repentance") patterns; 2. Experimental Findings: The proposed approach based on MARL shows remarkable detection performance in complex and dynamically evolving market environments. When facing various malicious bidding strategies, this method effectively improves the accuracy and robustness of malicious bidder detection. Compared with traditional detection methods based on rules or supervised learning, it has better adaptability in large-scale data processing scenarios and can more effectively capture malicious bidder behaviors in the market.

Subsequent sections of this study will focus on core methodology, experimental validation, and research prospects: Chapter 2 systematically reviews relevant technologies in malicious bidding detection, covering the current applications and limitations of machine learning, reinforcement learning, blockchain, and game theory, while highlighting the technical advantages of MARL. Chapter 3 proposes a MARL-based detection method. By constructing a multi-agent interaction framework and designing a dynamic adjustment mechanism for the maliciousness coefficient, it achieves precise modeling of bidding behavior and assessment of malicious intent; Chapter 4 validates the effectiveness of this method through simulation experiments, demonstrating superiority over traditional methods in detection accuracy, low false alarm rate, and scenario adaptability; Chapter 5 summarizes the research findings and identifies future directions for deepening the study, including optimizing learning algorithms, expanding multi-scenario validation, and addressing reward non-stationarity issues, thereby providing a feasible pathway for detecting malicious bidding in dynamic market environments.

## 2 Related work

With the rapid development of e-commerce and online auction markets, malicious bidding activities have increasingly disrupted market order. Traditional detection methods have gradually revealed limitations in addressing issues such as the covert nature, dynamic traits, and massive data volume of such behaviors. This section systematically reviews the current state of technology in malicious bidding detection, its core characteristics, and the shortcomings of existing methods, integrating relevant technical research and research background to lay the theoretical and practical foundation for proposing subsequent research methodologies.

Research on malicious bidding detection technology has spanned multiple fields, including machine learning, reinforcement learning, game theory, blockchain, and data mining (Ressi et al., 2024). As market environments grow more complex and malicious behaviors diversify, various technologies have been successively introduced to the field of malicious bidding detection, optimized through different methodologies. Existing approaches not only focus on enhancing detection accuracy but also continuously explore improvements to system adaptability and real-time responsiveness, as detailed in the following classifications. 1. Supervised Learning Algorithm Applications: Supervised learning algorithms train classification models using labeled data to effectively identify malicious bidding activities. However, this approach relies heavily on labeled data and faces challenges such as high data acquisition costs and inconsistent annotation quality–both of which limit its performance in practical applications; 2. Unsupervised Learning and Anomaly Detection Applications: In scenarios with scarce labeled data, unsupervised learning automatically identifies potential malicious bidding behaviors through techniques like clustering and anomaly detection (Zhang et al., 2020a). Nevertheless, unsupervised learning is sensitive to noisy data and lacks interpretability due to the absence of prior knowledge support. 3. Ensemble Learning and Optimization Approaches: To overcome the limitations of single algorithms, Zhou et al. employed ensemble learning, integrating multiple models including Support Vector Machines (SVM), Extreme Gradient Boosting (XGBoost), and k-Nearest Neighbors (k-NN) to construct a hybrid detection system, thereby enhancing overall detection performance. Concurrently, Mestiri (2024) proposed a "data quality—technical capability—business scenario adaptation" logic in credit scoring research, which offers insights for optimizing malicious bidding detection models by emphasizing the importance of data cleansing, dynamic feature extraction, and market-type adaptation (Majadi and Trevathan, 2024).

Notably, studies in related domains such as credit scoring (Mestiri, 2024) and financial fraud detection (Ma et al., 2019) have further demonstrated the effectiveness of machine learning in identifying anomalous patterns, providing cross-domain references for malicious bidding detection. Beyond basic machine learning, reinforcement learning has emerged as a significant research direction in this field in recent years. Unlike traditional machine learning, reinforcement learning enables agents to learn optimal strategies through continuous interaction with their environment, eliminating reliance on labeled data and allowing dynamic adaptation to market changes. For instance, Li et al. (2025) proposed a malicious bidding detection framework based on deep reinforcement learning: this approach processes market input information through deep neural networks and adjusts behavioral strategies based on reward mechanisms, achieving high detection accuracy in dynamic market environments. Similarly, Zhang et al. (2020b) applied multi-DQN reinforcement learning to EV charging bidding in electricity auction markets, verifying the effectiveness of reinforcement learning in dynamic bidding strategy adaptation. The versatility of MARL has been further verified in diverse domains beyond bidding systems. In electricity markets, Yin et al. (2025) leveraged MARL to optimize auction mechanisms, demonstrating its effectiveness in handling dynamic pricing and participant interactions. For strategic bidding scenarios, Wu et al. (2024) developed an intelligent bidding framework based on MARL, which adapts to real-time market changes to maximize participant benefits while maintaining fairness. In dynamic pricing, Qiao et al. (2024) proposed a distributed dynamic pricing model using MARL, enabling platforms to balance supply-demand relationships and user satisfaction. Beyond these, Yuan and Wang (2025) applied deep Q-networks (DQN) within a MARL framework to develop green computing solutions for sustainable supply chain management, expanding the practical scope of MARL

and confirming its adaptability to complex, multi-stakeholder environments–this strengthens our confidence in applying MARL to malicious bidding detection. Within the MARL framework, the dynamic maliciousness coefficient is not an independently evaluated parameter; Instead, it is deeply embedded within the "action selection—reward feedback—policy update" closed-loop system. Through mathematical modeling, this coefficient directly influences agent decision logic and learning processes (Dong and Finlay, 2025). Notably, the parameter adaptation logic of our dynamic maliciousness coefficient aligns with advanced optimization algorithms in related fields. The Improved Adaptive Differential Evolution (IADE) algorithm proposed by Zhou et al. (2021a) dynamically adjusts the scaling factor and crossover probability in differential evolution, effectively resolving parameter dependency and slow convergence in 6G network optimization—this provides a key reference for our control of maliciousness coefficient adjustment magnitude. Another multi-objective optimization algorithm, AFED-EF, also proposed by Zhou et al. (2021b), achieves energy-efficient virtual machine (VM) allocation in cloud data centers through fine-grained parameter tuning, further validating the effectiveness of "small-step adjustment" in maintaining system stability, which is consistent with our 0.001 fine-tuning increment design. Additionally, Zhou et al. (2018)'s earlier work on adaptive energy-aware algorithms for cloud data centers optimizes both SLA violation rates and power consumption, supporting the multi-objective coordination logic in our parameter adjustment–where we balance detection accuracy and reward stability.

Another critical technical integration in malicious bidding detection lies in the convergence of blockchain and machine learning. The decentralized and immutable nature of blockchain technology provides data security and transparency guarantees for detecting malicious bidding. Recording bidding activities on the blockchain enables traceability and verifiability of bidding history, reducing opportunities for malicious manipulation. Combined with smart contracts, blockchain technology can also automatically enforce bidding rules to prevent rule abuse (Ressi et al., 2024). For example, Hameed et al. integrated blockchain with machine learning models such as gradient-boosting decision trees (GBDT) and random forests to construct a fraud detection mechanism; Chaudhari and Malik proposed an advertising bidding system based on blockchain that integrates machine learning models–this system detects malicious behavior while ensuring transparent and traceable bidding processes through blockchain (Cao et al., 2017). Secure auction protocols also contribute to transparent bidding, for example, Zhou et al. (2024)'s efficient first-price sealed-bid electronic auction protocol under malicious models, which can complement our detection framework in ensuring bidding integrity.While MARL shows strong adaptability, reinforcement learning (RL) as a core technology still faces inherent challenges. Liao et al. (2025) pointed out that RL suffers from high computational complexity and lengthy training times, especially when processing large-scale datasets—this is also a key constraint for our MARL-based detection model, as massive historical bidding data may prolong training cycles. Addressing this issue, recent research has focused on optimizing RL's practical deployment: for example, Zhou and Abawajy (2025) proposed a reinforcement learning-based edge server placement strategy for intelligent Internet of Vehicles (IoV) environments, which enhances real-time responsiveness by reducing data transmission latency—this provides insights for our future optimization of real-time detection in high-frequency bidding scenarios. Currently, blockchain applications in this field face performance and scalability challenges: enhancing processing speed and ensuring efficient operation within high-frequency bidding systems remain unresolved issues. Beyond bidding systems, the integration of blockchain and AI has shown promise in enhancing data valuation frameworks (Theodorou and Theodorou, 2024), which may provide technical inspiration for optimizing malicious bidding detection systems. The intersection of game theory and cloud security also offers cross-domain references, such as Gill et al. (2024)'s systematic review of game-theoretic models for cloud security requirements, highlighting the potential of interdisciplinary integration in detection system optimization.

Complementing these technical approaches, game theory offers a crucial theoretical foundation for malicious bidding detection by treating bidding activities as a multi-party game process. By analyzing bidders' strategic choices, it infers potential malicious behavior (Sim, 2024). For instance, Li et al. (2025) proposed a malicious bidding detection model based on Nash equilibrium: this model constructs a game framework to analyze bidder strategy interactions and predict malicious behavior patterns. To further enhance detection effectiveness, integrating game theory with reinforcement learning has become a valuable direction. Gupta and Gupta (2023) combined both approaches to propose a novel malicious bidding detection framework–by simulating multi-agent strategy interactions and analyzing bidder behavior patterns, this framework demonstrates greater robustness and adaptability in complex markets, addressing some of the limitations of single-theory-based detection methods (Shi et al., 2025).

## 3 Research methodology

This paper proposes a malicious bidding detection method based on MARL. It models users under investigation as special agents. These agents learn optimal bidding strategies through game-theoretic interactions with adversarial agents, utilizing historical data. Additionally, this paper designs a dynamic maliciousness coefficient adjustment mechanism to accurately assess user maliciousness levels. Compared to traditional detection methods based on rules, or supervised learning, this approach demonstrates higher detection accuracy and adaptability in complex, dynamically changing market environments. Experimental results demonstrate that the multi-agent system effectively captures interactive relationships among bidders through game-based learning, dynamically adjusting detection strategies to significantly enhance the detection of malicious bidding behavior.

This design centers entirely on the maliciousness coefficient, a parameter used to preliminarily assess an individual's level of malicious bidding. The probability of a user becoming a genuine malicious bidder is positively correlated with their maliciousness

coefficient. The model described herein primarily involves two core steps: First, obtaining the initial maliciousness coefficient for the target user based on their historical data. Second, incorporating this initial coefficient into a multi-agent system. Notably, during subsequent detection processes, the maliciousness coefficients of user agents undergo dynamic adjustment to simulate a "repentance" process. The system architecture is illustrated in Figure 1.

The architecture diagram clearly illustrates the interaction process of a multi-agent system, which is divided into four steps:

Step 1: The system initializes the environment state and configures agent parameters.

Step 2: Agents perceive the environment, select actions, and receive rewards.

Step 3: The proxy updates its policies through learning and dynamically adjusts the maliciousness coefficient.

Step 4: The system obtains detection results and malicious user probabilities, then outputs data.

## 3.1 Calculation of the initial malicious coefficient

Let the historical auction dataset of the user to be detected be $D$, and one piece of historical data be *data*, that is, $data \in D$. The expression is $data = (bids, values, index)$, where *bids* are the bids of all participants in a certain bidding activity, *values* are the values of the bidding item in the minds of all participants, and index is the index subscript of the user to be detected. The calculation of the initial malicious coefficient is divided into four steps:

Step 1: Calculate the median bid (med) for each historical data point. Sort the historical bids and take the middle value as the median. Then, calculate the bid deviation $\theta$ for the target user using the following Equation 1.

$$\theta = \frac{\text{bids}_{\text{index}} - \text{med}}{\text{med}}, \tag{1}$$

Step 2: Introduce a function $f_i$, representing the probability that the target user becomes a malicious user based on historical data $i$. Specifically, as shown in Equation 2.

$$f_i = \begin{cases} 1, & \theta_i > \theta_0; \\ 0, & \text{otherwise.} \end{cases}, \tag{2}$$

Among these, $\theta_0$ represents the deviation threshold determined by the market, reflecting the market's tolerance for malicious overbidding, with a positive correlation between the two. $\theta_i$ denotes the bid deviation under historical data $i$. $f_i$ is a binary function indicating the probability that the user under investigation becomes a malicious user based on historical data $i$. When the bid deviation for historical data $i$ exceeds the market-determined deviation threshold–that is, when $f_i = 1$–it indicates the user has a probability of becoming a malicious user.

Step 3: Calculate the total sum of f for all data using Equation 3.

$$f_{\text{sum}} = \sum_{i=1}^{m} f_i, \tag{3}$$

Step 4: Let $\xi$ denote the maliciousness coefficient, which is used to preliminarily evaluate the degree of a user's malicious bidding (the probability of a user being a malicious bidder is positively correlated with $\xi$). Calculate the malicious coefficient $\xi$.

$$\xi = \frac{f_{sum}}{m}. \tag{4}$$

## 3.2 Multi-agent systems

A multi-agent system is a system composed of multiple agents, each possessing its own objectives and behavioral strategies. Agents interact with one another to achieve their respective goals. In this paper, the multi-agent system consists of a target user and multiple adversarial agents. The user is a special type of agent. This system is summarized by Equation 5.

$$system = (uesr, agent_2, agent_n). \tag{5}$$

Here, *user* represents the target user under investigation, and $agent_i$ denotes the $i-1$th adversarial agent. $n$ is the total number of agents. When locating agents via indexing, it is specified that index 1 corresponds to the target user, meaning indexing starts from 1.

The general process of multi-agent system operation is as follows: First, the target user and adversarial agents select their bids in each round of bidding according to their respective strategies. Next, the payoff for each agent is calculated based on the bid outcomes. Finally, agents adjust their strategies according to their payoffs. This process iterates continuously until the system reaches a Nash equilibrium state. The adaptability of multi-agent systems in complex optimization scenarios has been validated in other fields.

## 3.3 Action-value function in single-agent systems

Since the system discussed in this paper is a multi-agent system, and the action-value function for agents in a multi-agent system is based on the action-value function for a single agent, the action-value function for a single agent becomes particularly important. Therefore, we first introduce the action-value function for a single agent.

The action-value function $Q$ is defined as the expected reward an agent receives when taking a specific action in a given state. The agent's objective is to learn the action-value function and discover the optimal policy that maximizes the expected reward for each action taken in every state. Based on this definition, the expression for the action-value function is provided in Equation 6.

$$Q(s_t, a_t, \pi) = \mathbb{E}(U_t \mid S = s_t, A = a_t, \pi). \tag{6}$$

In Equation 6, $E$ denotes the expected value, where uppercase letters such as $S$ and $A$ represent random variables, while the lowercase expressions following the equals sign denote specific observed values. Here, $s_t$ represents the state at time $t$, $a_t$ represents the action of the agent at time $t$, $\pi$ represents the agent's strategy network, and its return value is the probability density of the action that the agent is going to take, that is $a \sim \pi$. $Q$ represents the

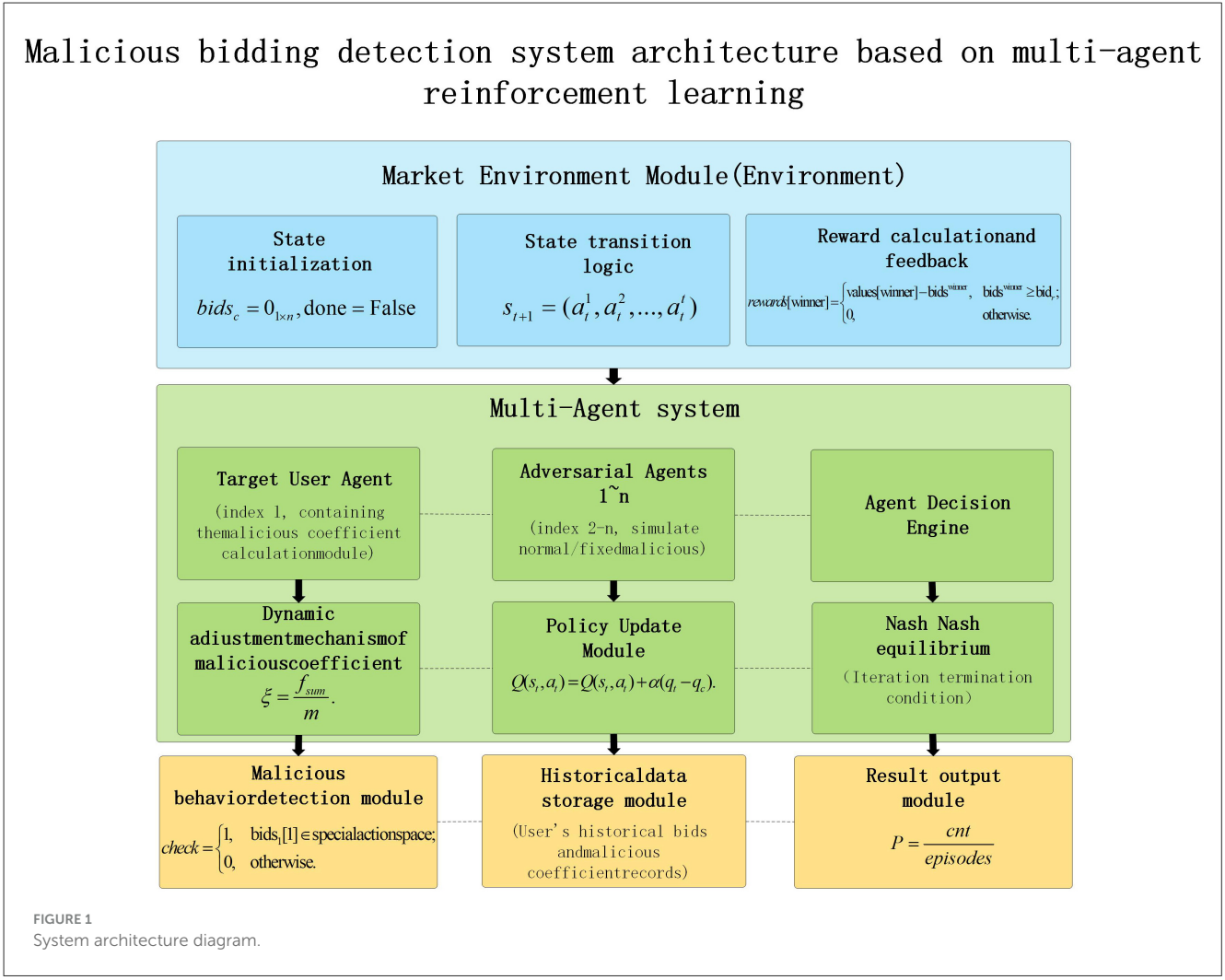


FIGURE 1
System architecture diagram.

action-value function, which has three independent variables $s_t$, $a_t$ and $\pi$.

$U_t$ denotes the agent's reward at time $t$, with its specific expression shown in Equation 7.

$$U_t = \sum_{j=0}^{\infty} \gamma^j r_{t+j}. \tag{7}$$

In Equation 7, $r_t$ denotes the reward received by the agent at time step $t$, while $\gamma$ represents the discount factor. The discount factor serves to discount future rewards, thereby enabling these future rewards to influence the current action-value function. Incorporating the discount factor also stabilizes the action-value function, preventing the emergence of infinite returns. On the other hand, it encourages the agent to focus more on immediate rewards rather than future ones.

The optimal action value function $Q^*$ guides the agent to take the most desirable actions. Its relationship with the action value function is shown in Equation 8.

$$Q^*(s_t, a_t) = \max_{\pi} Q(s_t, a_t, \pi). \tag{8}$$

From Equation 8, it follows that the optimal action value function $Q^*$ is the maximum value of the action value function $Q$ across all policy networks $\pi$.

## 3.4 Agent's decision-making process

The agent's action can be obtained from Equation 9.

$$a = \begin{cases} \text{randomchoice(normal\_action\_space)}, & rand < \epsilon, rand \geq \xi; \\ \text{randomchoice(spite\_action\_space)}, & rand < \epsilon, rand < \xi; \\ \arg\max_{a \in \text{normal\_action\_space}} Q^*(s, a), & rand \geq \epsilon, rand \geq \xi; \\ \arg\max_{a \in \text{special\_action\_space}} Q^*(s, a), & rand \geq \epsilon, rand < \xi. \end{cases} \tag{9}$$

Where *randomchoice* means randomly selecting an element from a set passed as its argument, rand returns a randomly-generated decimal within the range of [0, 1], $\epsilon$ is a hyper-parameter set to allow the agent the opportunity for free exploration, $\xi$ is the agent's malicious coefficient, *normalactionspace* is the normal action space of the agent, and *spiteactionspace* is the malicious action space of the agent.

When "$rand \geq \epsilon$", the action $a$ obtained from the above formula may be a null value. At this point, action $a$ is obtained through Equation 10.

$$a = \begin{cases} \text{randomchoice(normalactionspace)}, \text{rand} \geq \xi; \\ \text{randomchoice(spiteactionspace)}, \text{rand} < \xi. \end{cases} \tag{10}$$

## 3.5 Optimal action-value function of the agent

We use $Q$-Learning to learn the agent's strategy and obtain $Q^*$ through this algorithm.

Step 1: Obtain the current state $s_t$ and action $a_t$. Use Equation 11 to calculate the current action value function $q_c$.

$$q_c = Q(s_t, a_t), \tag{11}$$

Step 2: Approximately obtain the optimal action-value function $q_r$ through the Temporal Difference (TD) and Monte Carlo algorithms; the algorithmic expression is shown in Equation 12.

$$q_r = r_t + \max Q(s_{t+1}, a_{t+1}), \tag{12}$$

Step 3: Update the action-value function using the gradient ascent method.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(q_t - q_c). \tag{13}$$

Here, $\alpha$ is the learning rate.

Updating the action-value function using the gradient ascent method means updating the $Q$-table with this algorithm. Here, $q_c$ is the action value generated by the current action-value function and can be obtained from the $Q$-table. $q_r$ is the action value under the optimal action-value function and is approximately obtained through the TD and Monte Carlo algorithms.

It should be noted that the dynamic adjustment of the maliciousness coefficient leads to non-stationarity in the reward function. Since reward calculations are directly linked to the real-time maliciousness coefficient, the agent's reward feedback fluctuates over time as our method dynamically adjusts this coefficient based on user behavior. This disrupts the convergence assumption of "fixed reward function" in traditional MARL. This non-stationarity may have two implications: First, the agent's action value function (Q-function) updates may exhibit oscillations. For instance, a reduction in the maliciousness coefficient in one round may weaken reward penalties, only for the coefficient to rebound in the next round and intensify penalties, potentially causing Q-values to fluctuate repeatedly near optimal values. Second, it increases the difficulty of strategy coordination among multiple agents. The policy updates of ordinary adversarial agents depend on the maliciousness coefficient of the "agent" agent. If the coefficient is adjusted too frequently, ordinary agents struggle to adapt quickly to reward changes, potentially slowing the system's convergence to Nash equilibrium.

TABLE 1 Core experimental parameters.

| Param. cat. | Param. name | Val. | Desc. |
|---|---|---|---|
| Training | *EPISODE_NUM* | 5,000 | MARL→ Nash |
| | $\alpha$ | 0.01 | Q-update step |
| | $\gamma$ | 0.9 | Reward weight |
| | $\epsilon$ | 0.1 | Exploration prob. |
| Env. | *NUM_AGENTS* | 3(1+2) | 1 target+2 adv. |
| | *MIN_AUC_PR* | 5 | Min bid (S5.1) |
| | *MAX_NOR_BID* | 10 | Non-malicious limit |
| | *AUC_T* | 10 | Max steps/auction |
| Test | *TEST_NUM* | 100 | Samples/$\xi$ |
| | *TEST_RUN_NUM* | 50 | Reduce error |
| | $\theta_0$ | 0.2 | Abnormal judge (S4.1) |

# 4 Experiments

## 4.1 Environment

The environment is set as a first-price sealed-bid auction. In this environment, the action of an agent is to place a bid. Here, $\text{bids}_c$ represents the current bids of multiple agents, $\theta_0$ is the market deviation coefficient, $\text{bid}_r$ is the minimum successful auction price, also known as the reserve price. That is, only when the $\max(\text{bids}_c) \geq \text{bid}_r$ does it represent a successful auction. The basic experimental parameters are set as shown in Table 1.

### 4.1.1 Environment initialization

Set the current bid vector as a row vector of all zeros, and set the status of whether the auction is over to "No".

$$\text{bids}_c = 0_{1\times n}, \text{done} = \text{False}, \tag{14}$$

The number of agents in the multi-agent system is $n$. Therefore, the current bid vector $bids_c$ is initialized as a $1 \times n$ zero vector.

### 4.1.2 Environment state transition

The function corresponding to the environment state transition is *step*. This function can be regarded as a process of transforming from one state to another. Its return values are the next state, the agent's reward, and whether the episode is over.

The implementation of this function is divided into two steps: 1. State transition and judging whether the auction is over; 2. Reward setting.

Step 1: Specific operations of state transition and judging whether the auction is over. In this environment, if the highest bid is positive, the auction can end.

$$done = \begin{cases} \text{True}, max(bid_c) > 0 \\ \text{False}, otherwise \end{cases}, \tag{15}$$

It should be noted that the state of the environment at the next moment is the bids of all agents at the current moment. That is:

$$s_{t+1} = (a_t^1, a_t^2, ..., a_t^t), \tag{16}$$

Here, $a^i$ represents the action of the $i$-th agent at time t. It is worth noting that the initial state is the initial bid vector.

Step 2: Specific operations for *rewards* setting:

1. Initialize the reward vector rewards:

$$rewards = o_{1\times n}, \tag{17}$$

2. Select the agents whose bids are greater than the median bid to form the first agent index vector $vector^1$:

$$\text{vector}_{\text{idx}}^1 = \{i \mid \text{bids}_i > \text{bids}_{\text{med}}\}, \tag{18}$$

3. Calculate the bid deviation $\theta_i$ of the corresponding agents in $vector_1$:

$$\theta = \frac{\text{bids}_i - \text{bids}_{\text{med}}}{\text{bids}_{\text{med}}}, \tag{19}$$

Here, $bids_{med}$ is the median bid, and $\theta_i$ is the bid deviation of the agent with index $i$.

4. Select the indices of the agents that satisfy $\theta_i > \theta_0$, where $i \in \text{vector}_{\text{idx}}^1$, to form the second agent index vector $\text{vector}_{\text{idx}}^2$:

$$\text{vector}_{\text{idx}}^2 = \{\theta_i > \theta_0 \mid i \in \text{vector}_{\text{idx}}^1\}, \tag{20}$$

5. Give negative rewards, also known as excessive-deviation rewards, to the agents corresponding to $\text{vector}_{\text{idx}}^2$:

$$\text{reward}[i] \leftarrow \text{reward}[i] - \frac{\theta_i \times \text{value}[i]}{\theta_{\text{sum}} \times \theta_0}, \quad i \in \text{vector}_{\text{idx}}^2, \tag{21}$$

Where, $\theta_{\text{sum}} = \sum_{i\in\text{vector}_{\text{idx}}^2} \theta_i$

6. Determine whether the auction is over. If the auction is over, issue the winner's reward:

$$rewards[\text{winner}] = \begin{cases} \text{values}[\text{winner}] - \text{bids}^{\text{winner}}, & \text{bids}^{\text{winner}} \geq \text{bid}_r; \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

That is to say, only when the winner's bid is not less than the reserve price can the winner win the auction and obtain the auction item. Here, $winner = \arg\max(bids_c)$.

Return Value Description: The *step* function returns $(s_{t+1}, rewards, done)$.

## 4.2 Training

To make the multi-agent system reach the Nash equilibrium state, the multi-agent system is trained. The specific training steps are as follows:

Step 1: Obtain the malicious coefficient $\xi$ through historical data and initialize the malicious coefficients of all agents.

Step 2: Start the training loop, and each loop is called an episode. The specific operations within an episode are as follows:

1. Initialize the environment and obtain the initial state $s$.

2. Initialize the episode-end flag *done*.

3. When the episode is not over, perform the following operations.

4. Obtain the actions $a$ of all agents.

5. Use the *step* function to obtain the next state $s_{next}$, the rewards *rewards* of all agents, and the episode-end flag *done*.

6. Update the strategies of the agents through the agents' action-value function $Q$.

7. Update the state $s$ to $s_{next}$.

8. When the episode ends, move on to the next episode.

Step 3: After the training is completed, return the trained multi-agent system.

## 4.3 Detecting malicious behaviors

The function corresponding to detecting malicious behaviors is the *check* function, and its target is the user to be detected, that is, Agent No. 1.

The specific function expression is as follows:

$$check = \begin{cases} 1, & \text{bids}_1[1] \in \text{specialactionspace}; \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

```
Input: ξ, agents, env
Output: agents_trained
for each agent ∈ agents do
  agent.ξ ← ξ
end for
for each episode ∈ [1, episodes] do
  state ← env.reset()
  done ← False
  while done = False do
    actions ← NULL
    for each agent ∈ agents do
      action ← agent.getaction(state)
      actions ← actions + action
    end for
    state_next, rewards, done ← env.step(actions)
    index ← 1
    for each agent ∈ agents do
      agent.learn(state, actions[index], rewards[index], state_next)
      index ← index + 1
    end for
    state ← state_next
  end while
end for
return agents
```

Algorithm 1. Training multi-agent systems.

## 4.4 Dynamic adjustment of the malicious coefficient

To simulate the "repentance" of the user agent to be detected, the malicious coefficient of this agent can be dynamically adjusted. If the user agent does not bid excessively high and the malicious coefficient remains positive after adjustment, the malicious coefficient can be fine-tuned. The adjustment conditions and methods are as follows:

$\text{agents}[1].\xi \leftarrow \text{agents}[1].\xi - 0.001$

$\text{agents}[1].\xi > 0.001, \theta_1 \leq \theta_0$

## 4.5 Obtaining the probability of a malicious user

Use the trained multi-agent system to test the user to be detected and obtain the probability that the user is a malicious one. During the testing process, the malicious coefficient of the user agent will be dynamically adjusted to simulate the "repentance" process.

The specific operations of the testing process are as follows:

Step 1: Obtain the multi-agent system $agents_{trained}$ based on the trained multi-agent system and the environment *env*.

Step 2: Initialize the counter *cnt*.

Step 3: Start multiple simulated auction processes, and each simulation is called an episode. The specific operations within an episode are as follows:

1. Initialize the environment and obtain the initial state *s*.

2. Initialize the episode-end flag *done*.

3. When the episode is not over, perform the following operations:

4. Obtain the actions *a* of all agents.

5. Use the *step* function to obtain the next state $s_{next}$, the rewards *rewards* of all agents, and the episode-end flag *done*.

6. Update the state *s* to $s_{next}$.

7. When the episode ends, check if the malicious coefficient of the user agent to be detected is greater than 0.001. If it is greater than 0.001 and the bid deviation $\theta_1$ is less than or equal to $\theta_0$ , fine-tune its malicious coefficient.

8. Add the return value of the *check* function to the counter *cnt*.

Step 4: Calculate the probability of a malicious user: $P = \frac{cnt}{episodes}$.

Step 5: Return the probability *P* that the user is a malicious one. The pseudocode of the testing process is as follows:

**Input:** $\xi$, $\text{agents}_{\text{trained}}$, env
**Output:** Malicious user probability $P$
Initialize counter $cnt \leftarrow 0$
**for** each episode $\in [1, \text{test\_episodes}]$ **do**
  state $\leftarrow$ env.reset()
  done $\leftarrow$ False
  **while** done = False **do**
    actions $\leftarrow$ NULL
    **for** each agent $\in \text{agents}_{\text{trained}}$ **do**
      action $\leftarrow$ agent.getaction(state)
      actions $\leftarrow$ actions + action
    **end for**
    $\text{state}_{\text{next}}$, rewards, done $\leftarrow$ env.step(actions)
    state $\leftarrow \text{state}_{\text{next}}$
  **end while**
  **if** $\text{agents}_{\text{trained}}[1].\xi > 0.001$ AND $\theta_1 \leq \theta_0$ **then**
    $\text{agents}_{\text{trained}}[1].\xi \leftarrow \text{agents}_{\text{trained}}[1].\xi - 0.001$
  **end if**
  $cnt \leftarrow cnt + \text{check}(\text{agents}_{\text{trained}}[1])$
**end for**
$P \leftarrow \frac{cnt}{\text{test\_episodes}}$
**return** $P$

Algorithm 2. Testing malicious user probability.

TABLE 2 $\xi$ vs. malicious probability.

| $\xi$ | P(%) ± CI | Behavior |
|---|---|---|
| 0.1 | 12.3 ± 1.5/[9.4,15.2] | Extreme low |
| 0.2 | 23.7 ± 2.1/[19.6,27.8] | Low (10%-30%) |
| 0.3 | 35.2 ± 2.4/[30.5,39.9] | Moderate (tentative) |
| 0.4 | 46.8 ± 2.7/[41.5,52.1] | Moderate (50%) |
| 0.5 | 58.5 ± 2.9/[52.8,64.2] | High (>50%) |
| 0.6 | 69.3 ± 3.1/[63.2,75.4] | High (70%) |
| 0.7 | 78.6 ± 3.3/[72.1,85.1] | Extreme high (frequent) |
| 0.8 | 86.4 ± 3.5/[79.5,93.3] | Extreme high (almost all) |
| 0.9 | 92.1 ± 3.2/[85.8,98.4] | Stable (near 100%) |
| 1.0 | 96.7 ± 2.8/[91.2,100.0] | Fully malicious |

Fitting equation: $P = 98.2\xi - 2.5$ (coefficient of determination $R^2 = 0.987$), all standard deviations < 3.5%.

## 4.6 Experimental results

By deriving corresponding maliciousness coefficients from extensive historical data, we train a multi-agent system. Subsequently, through multiple simulated auction processes, we obtain the probability of malicious behavior for the users under detection. During the experiment, we set the following parameters: learning rate: 0.01; discount factor: 0.9; exploration rate: 0.1; training rounds: 5000; number of bidders: 3; minimum selling price: 5; maximum normal bid: 10; auction cycle: 10; test runs: 100; initial malicious bid probability: 0.95; test runs per batch: 50; and malicious bid threshold: 1.0. Table 2 shows the quantitative relationship between the maliciousness coefficient and the probability of malicious behavior.

Figure 2 shows the maliciousness coefficient on the horizontal axis and the maliciousness probability on the vertical axis. The graph reveals a positive correlation between the two. When the maliciousness coefficient is low, the latter is also low; Conversely, when the former is high, the maliciousness probability increases. This indicates that the maliciousness coefficient effectively reflects the malicious level of the users under detection, and thus provides a valuable reference for identifying the malicious users.

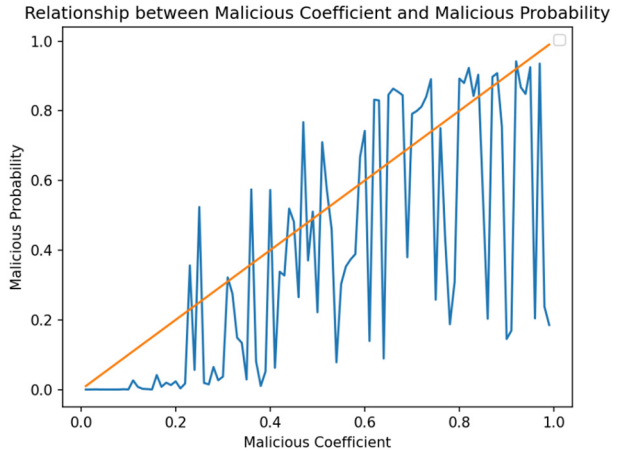


FIGURE 2
Relationship between malicious coefficient and malicious probability.

## 5 Conclusions and future work

In the field of malicious bidder detection, this paper explores a novel technical approach by introducing Multi-Agent Reinforcement Learning (MARL) to construct detection methods. By establishing a multi-agent system, the study simulates bidding interactions within market environments and dynamically optimizes the setting of maliciousness coefficients. This ultimately achieves effective identification of malicious users. Experimental validation demonstrates that this approach exhibits superior accuracy and robustness compared to traditional detection schemes relying on rule-based judgments or supervised learning when confronting complex, dynamically evolving market scenarios. Simultaneously, this research addresses two critical challenges in practical applications: first, it provides a viable solution for scenarios with scarce data samples and inaccurate labeling; Second, it explores potential "repentance" behaviors exhibited by malicious users, thereby enhancing the integrity of detection logic. Overall, this MARL-based technology offers a novel direction for detecting malicious bidding by overcoming limitations of existing methods and achieving innovation through two core capabilities: First, its built-in reward-punishment mechanism enables autonomous adjustment of detection strategies, allowing flexible adaptation to dynamic market environments; Second, it precisely captures the interactive relationships among different bidders, enabling more realistic predictions of their subsequent behavioral patterns and laying the foundation for enhanced detection effectiveness.

In the specific design of this multi-agent framework, modeling target users as "Agent" entities requires achieving dual objectives: "behavioral simulation" and "malicious attribution". The former involves replicating normal bidding habits, while the latter focuses on dynamically adjusting malicious coefficients to reflect evolving intent. This design overcomes the "Homogeneous Agent" limitation of traditional multi-agent systems, serving as a critical bridge between historical data features and RTB behavior. Simultaneously, the multi-agent system's "dynamic perception-feature transfer" capability overcomes limitations imposed by known malicious patterns—even when encountering previously unseen collusive or adversarial manipulations, it can progressively identify them through feature matching. This addresses the traditional model's "effective against known strategies, ineffective against unknown strategies" problem.

Building on the current framework, future research in related fields may expand in several key directions. First, further optimization of agent learning algorithms could improve detection efficiency and model convergence speed, while simulation and strategy combinations can be employed to further enhance the model's adaptability to diverse scenarios. Second, given the complexity and diversity of real-world market environments and bidding strategies, consideration should be given to incorporating more diverse settings. A key technical challenge for future research lies in addressing reward non-stationarity caused by dynamic adjustment of adversarial coefficients. To mitigate this issue, optimization approaches can focus on two aspects: exploring adaptive adjustment frequencies and introducing specialized optimization algorithms tailored for non-stationary MARL problems. These optimization methods not only alleviate the impact of non-stationarity on convergence but also enhance policy stability during long-term operation. While the model tested in this paper primarily targets a single market environment, future validation across markets or in more complex multi-market settings could assess its universality. Furthermore, integrating the framework with real-time monitoring systems to achieve live malicious detection represents another promising avenue for exploration.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

PW: Writing – original draft, Writing – review & editing, Conceptualization. YW: Writing – review & editing, Visualization. YZ: Writing – original draft, Data curation. YL: Writing – original draft. ZH: Writing – original draft. DT: Writing – original draft. YL: Writing – original draft.

## Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This research was financially supported by the Natural science foundation of Fujian Province (Grant No. 2024J08372).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

## Publisher's note



## References

Cao, L. (2022). Ai in finance: Challenges, techniques, and opportunities. *ACM Comp. Surv*. 55, 1–38. doi: 10.1145/3502289

Cao, S., Yang, X., Zhou, J., Li, X., Qi, Y., and Xiao, K. (2017). "Poster: Actively detecting implicit fraudulent transactions," in *Proceedings of the 02017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

Chen, C., Wang, G., Liu, B., Song, S., Mao, K., Yu, S., et al. (2024). Real-time bidding with multi-agent reinforcement learning in multi-channel display advertising. *Neural Comp. Appl.* 37, 499–511. doi: 10.1007/s00521-024-10649-6

Dong, S. C., and Finlay, J. R. (2025). Dynamic reinsurance treaty bidding via multi-agent reinforcement learning. *arXiv* [preprint] arXiv:2506.13113. doi: 10.48550/arXiv.2506.13113

Gill, K. S., Sharma, A., and Saxena, S. (2024). A systematic review on game-theoretic models and different types of security requirements in cloud environment: challenges and opportunities. *Arch. Comp. Methods Eng.* 31, 3857–3890. doi: 10.1007/s11831-024-10095-6

Guo, H., Zhao, X., Yu, H., Zhang, X., and Li, J. (2021). Game analysis of merchants and consumers confronting fakes on e-commerce platforms. *Syst. Sci. Cont. Eng.* 9, 198–208. doi: 10.1080/21642583.2021.1891992

Gupta, R., and Gupta, J. (2023). Federated learning using game strategies: State-of-the-art and future trends. *Comp. Netw*. 225:109650. doi: 10.1016/j.comnet.2023.109650

Kanmaz, M., and Surer, E. (2020). Using multi-agent reinforcement learning in auction simulations. *arXiv* [preprint] arXiv:2004.02764. doi: 10.48550/arXiv.2004.02764

Li, H., Yang, P., Liu, W., Yan, S., Zhang, X., and Zhu, D. (2025). Multi-agent reinforcement learning in games: Research and applications. *Biomimetics* 10:375. doi: 10.3390/biomimetics10060375

Liao, Q., Feng, Z., Wu, H., Chen, S., and Xue, X. (2025). Self-organization scheduling of dynamic aav-md networks via three-layer hierarchical reinforcement learning. *IEEE Trans. Consum. Elect.* 71:3790–3801. doi: 10.1109/TCE.2025.3544802

Ma, T.-S., Qian, S., Cao, J., Xue, G., Yu, J., Zhu, Y., et al. (2019). "An unsupervised incremental virtual learning method for financial fraud detection," in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications*, 20190.

Majadi, N., and Trevathan, J. (2024). Detecting collusive seller shill bidding using social network analysis. *J. Inform. Syst. Secur.* 20:111–139.

Mestiri, S. (2024). Credit scoring using machine learning and deep learning-based models. *Data Sci. Finance Econ*. 4:236–248. doi: 10.3934/DSFE.2024009

Mo, K., Ye, P., Ren, X., Wang, S., Li, W., and Li, J. (2024). Security and privacy issues in deep reinforcement learning: threats and countermeasures. *ACM Comp. Surv*. 56, 1–39. doi: 10.1145/3640312

Qiao, W., Huang, M., Gao, Z., and Wang, X. (2024). Distributed dynamic pricing of multiple perishable products using multi-agent reinforcement learning. *Expert Syst. Appl.* 237:121252. doi: 10.1016/j.eswa.2023.121252

Ressi, D., Romanello, R., Piazza, C., and Rossi, S. (2024). AI-enhanced blockchain technology: a review of advancements and opportunities. *J. Netw. Comp. Appl.* 225:103858. doi: 10.1016/j.jnca.2024.103858

Shi, R., Liu, Y., Ying, X., Tan, Y., Feng, Y., Ai, L., et al. (2025). Hide-and-shill: A reinforcement learning framework for market manipulation detection in symphony-a decentralized multi-agent system. *arXiv* [preprint] arXiv:2507.09179. doi: 10.48550/arXiv.2507.09179

Sim, K. M. (2024). A strongly group strategyproof and shill resistant bargaining mechanism for fog resource pricing. *Dynam. Games Appl.* 14, 1238–1267. doi: 10.1007/s13235-023-00550-7

Theodorou, P., and Theodorou, T. (2024). Valuation of big data analytics quality and competitive advantage with strategic alignment model: from greek philosophy to contemporary conceptualization. *Data Sci. Finance Econ.* 4, 53–64. doi: 10.3934/DSFE.2024002

Wu, J., Wang, J., and Kong, X. (2024). Intelligent strategic bidding in competitive electricity markets using multi-agent simulation and deep reinforcement learning. *Appl. Soft Comput*. 152:111235. doi: 10.1016/j.asoc.2024.111235

Yin, B., Weng, H., Hu, Y., Xi, J., Ding, P., and Liu, J. (2025). Multi-agent deep reinforcement learning for simulating centralized double-sided auction electricity market. *IEEE Trans. Power Syst*. 40, 518–529. doi: 10.1109/TPWRS.2024.3404472

Yuan, D., and Wang, Y. (2025). Sustainable supply chain management: A green computing approach using deep q-networks. *Sustain. Comp.: Inform. Syst*. 45:101063. doi: 10.1016/j.suscom.2024.101063

Zhang, T., Ye, D., Zhu, T., Liao, T., and Zhou, W. (2020a). Evolution of cooperation in malicious social networks with differential privacy mechanisms. *Neural Comp. Appl.* 35, 12979–12994. doi: 10.1007/s00521-020-05243-5

Zhang, Y., Zhang, Z., Yang, Q., An, D., Li, D., and Li, C. (2020b). Ev charging bidding by multi-dqn reinforcement learning in electricity auction market. *Neurocomp*. 397, 404–414. doi: 10.1016/j.neucom.2019.08.106

Zhou, D.-W., Zhou, D.-W., Chen, S.-Z., Cao, S.-Z., Zhao, X., Xing, D.-D., et al. (2024). Efficient first-price sealed e-auction protocol under secure multi-party computational malicious model. *J. Inform. Sci. Eng.* 35, 065–081. doi: 10.53106/199115992024023501005

Zhou, Z., and Abawajy, J. (2025). Reinforcement learning-based edge server placement in the intelligent internet of vehicles environment. *IEEE Trans. Intellig. Transport. Syst*. 2025, 1–11. doi: 10.1109/TITS.2025.3557259

Zhou, Z., Abawajy, J., Chowdhury, M., Zhigang, H., Keqin, L., et al. (2018). Minimizing sla violation and power consumption in cloud data centers using adaptive energy-aware algorithms. *Future Generat. Comp. Syst*. 86:836–850. doi: 10.1016/j.future.2017.07.048

Zhou, Z., Shojafar, M., Abawajy, J., and Bashir, A. K. (2021a). IADE: An improved differential evolution algorithm to preserve sustainability in a 6G network. *IEEE Trans. Green Commun. Network*. 5, 1747–1760. doi: 10.1109/TGCN.2021.3111909

Zhou, Z., Shojafar, M., Alazab, M., Abawajy, J., and Li, F. (2021b). Afed-ef: An energy-efficient vm allocation algorithm for iot applications in a cloud data center. *IEEE Trans. Green Commun. Network*. 5, 658–669. doi: 10.1109/TGCN.2021.3067309