

Research Article

IIDQN: An Incentive Improved DQN Algorithm in EBSN Recommender System

Jianan Guo ¹, Yilei Wang ^{1,2}, Hui An,¹ Ming Liu,¹ Yiting Zhang,¹ and Chunmei Li¹

¹School of Computer Science, Qufu Normal University, Rizhao, China

²Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, China

Correspondence should be addressed to Yilei Wang; wang_yilei2019@qfnu.edu.cn

Received 27 August 2021; Revised 4 October 2021; Accepted 25 March 2022; Published 17 May 2022

Academic Editor: Yuling Chen

Copyright © 2022 Jianan Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Event-based Social Networks (EBSN), combining online networks with offline users, provide versatile event recommendations for offline users through complex social networks. However, there are some issues that need to be solved in EBSN: (1) The online static data could not satisfy the online dynamic recommendation demand; (2) the implicit behavior information tends to be ignored, reducing the accuracy of the recommendation algorithm; and (3) the online recommendation description is inconsistent with the offline activity. To address these issues, an Incentive Improved DQN (IIDQN) based on Deep Q-Learning Networks (DQN) is proposed. More specifically, we introduce the agents to interact with the environment through online dynamic data. Furthermore, we consider two types of implicit behavior information: the length of the user's browsing time and the user's implicit behavior factors. As for the problem of inconsistency, based on blockchain technology, a new activities event approach on EBSN is proposed, where all activities are recorded on the chain. Finally, the simulation results indicate that the IIDQN algorithm greatly outperforms in mean rewards and recommendation performance than before DQN.

1. Introduction

EBSN, event-based social networks, is a new type of social network, which connects strangers through online events recommendation. These events and activities enrich the users' experience of offline activities and broaden the social scope of users. Namely, individual interest needs could be satisfied with EBSN, so that everyone can sponsor activities offline or participate in other people's activities based on their interests, such as language learning, sports, travel, reading, etc. Therefore, EBSN expand individuals' social network. Correspondingly, with the continuous development of big data and artificial intelligence technologies, online network recommendations and evaluation feedback are becoming more influential on offline social activities. However, the current recommendation needs of EBSN could not be satisfied by traditional recommendation system technology. Liao [1] pointed out three main challenges in EBSN: existing recommendation algorithms cannot respond to the event evaluation feedback due to the lack of display

preferences in the EBSN; the data-sparse problem is still severe; and the description of activity events in EBSN is complex and diverse, with high-dimensional requirements for preferences. That is to say, considering some implicit information can increase the probability of a more accurate recommendation. Accordingly, reinforcement learning for the recommendation is considered to be applied in EBSN recommendations.

1.1. Recommendation System and Reinforcement Learning. DQN serves as an off-policy strategy combining the neural network in deep learning with the Q-learning algorithm in reinforcement learning. The Google DeepMind team first published a paper on playing Atari with deep reinforcement learning in 2013 [2, 3]. In this paper, deep learning was for the first time linked with reinforcement learning. Q-learning algorithm [4] (the Q-learning algorithm uses the maximum Q value in the Q table to select the action with the best future return, Q-table consists of all states s and all actions a in the

current state s) and neural networks are applied to calculate the Q value (The Q value is used to evaluate the value of the agent's choice of action a in state s). At the same time, the experience replay buffer is used to solve uneven data distribution. Integrating reinforcement learning and artificial neural networks allows the machine to learn from its previous experience and improve continuously. Therefore, these experiences learned through deep reinforcement learning can learn more strategies that users cannot capture under hidden behaviors. Since the recommendation of the event networks is often related to the user's activities in the recent period, static view recommendations are used in most current recommendation models, which ignore the fact that event recommendation is a dynamic sequential decision process. To overcome this drawback, reinforcement learning is applied as an attempt to recommend activities. Wu X [5] points out that compared with traditional collaborative filtering algorithms, reinforcement learning algorithms can not only easily handle the problem of large discrete state action data but also take into account the impact of users' real-time data changes. Therefore, recommendation models based on reinforcement learning are increasingly being studied by people in recent years.

1.2. Related Work. The current recommendation methods using deep learning in EBSN can discover the potential feature information in the recommendation, turning specific features into abstract features. Recently, the research of reinforcement learning in recommendation algorithms mainly includes two aspects: One is based on the input data of the recommendation system, which is divided into methods using user content information [6, 7] and methods not using user content information [8]; the other is based on the output data of the recommendation system, predicting the method of item ranking, respectively [9, 10], and the method of predicting the user scoring of items [11, 12]. Wang and Tang [13] constructed an Event2Vec model using spatial-temporal information to optimize the recommendation in EBSN. Wang et al. [14] used CNN with word embedding to capture the contextual information in EBSN but only used word embedding without considering the recommendation impact of other factors. Luceri et al. [15] used the DNN framework to predict social behavior in EBSN. We incorporate these algorithms into the consideration of the recommended results and tested them in experiments.

However, although the above solutions have been significantly improved in terms of recommendation in EBSN, there are still better solutions to optimize these algorithms, such as reinforcement learning. There are many applications of the DQN algorithm of reinforcement learning in the recommendation. Chen [16] uses a value-based DQN algorithm to recommend tips, but he only uses the keywords in the search as the feature value. He does not take into account the impact of other features like hidden features on the recommendation. Zheng [17] uses DQN to construct a DR-based IRS for news recommendations. Similarly, in another DQN-based IRS proposed by Zhao et al. [18], two

separate RNNs capture sequential positive and negative feedback. However, value-based models are not easy to handle when the action state space is vast [19].

With the continuous development of blockchain technology, various technologies in the blockchain provide a comprehensive guarantee for the security of large complex heterogeneous networks. We consider a variety of data security and integrity technologies: such as encryption mechanisms to ensure data integrity [20–23]. Defend and analyze security from the perspective of game theory [24]. Therefore, the use of blockchain technology in the EBSN network is a good way to ensure the overall security in EBSN.

1.3. Motivations and Contributions. There are some issues that need to be addressed in the recommendation system, such as a lack of dynamic recommendation, the ignorance of user's implicit behavior information as well, as the urgent need to improve online data security (e.g. inconsistency). In this paper, we focus on the above three issues. Firstly, the DQN algorithm in the reinforcement learning is introduced into the recommendation, and applied in the online activity recommendation by using the good interaction between the agent and the environment. In the proposed algorithm, the agent is regarded as the recommendation system, and the interaction process between the user and the recommendation system is regarded as the interaction process between the agent and the environment, which reflects the dynamic characteristics of the recommendation process. This avoids the drawbacks of traditional recommendations that only rely on user historical data recommendations. Secondly, in order to reflect the effects of implicit information on the recommendation algorithm, we introduced the concept of time parameters. Assign a value to the interest of a certain activity information based on the user's browsing time to identify the real intention of the user's browsing activity, thereby removing irrelevant data from the sample data. Finally, Blockchain is introduced to provide a mechanism due to the urgent requirement of data consistency. This mechanism guarantees the accuracy of online recommendations by constraining the event sponsor's event descriptions in an honest and reliable way, thus promoting the organic combination of online recommendations and offline activities.

The main contributions are as follows:

- (1) The idea of a deep Q-networks algorithm in reinforcement learning is applied to the recommendation problem of EBSN to avoid the problems of sparse matrix and poor interaction in traditional networks. Furthermore, compared with existing methods, experiments' results show that the recommendation algorithm using reinforcement learning can get higher rewards than other recommendation algorithms.
- (2) Considering the user's implicit interest, an IIDQN algorithm is proposed to improve the DQN algorithm from two perspectives: Identifying hidden nodes in the neural networks which represent implicit interest, and incentivizing those hidden nodes;

a parameter related to browsing time is added in the reward calculation, and the users' interest in the activity is explicitly demonstrated by the browsing time. Experiments' results show that the mean reward and accuracy obtained by IIDQN are significantly better than those of the DQN algorithm.

- (3) A framework is proposed based on blockchain to ensure honest behaviors for each user in the EBSN networks. This framework restricts all members in the event networks to publish "honest" offline activities in accordance with the activity information.

2. An Example of the Inactive Improved DQN

For the event recommendation part of event participants, we discuss the following issues and clarify that using reinforcement learning for event recommendation could easily infer some hidden behaviors of users. The following examples are precedent descriptions of the problems in the EBSN recommendation.

2.1. Finding Hidden Information Points of User Implicit Behaviors in Recommendation. We take the social event in Tokyo, Japan, in the Meetup as an example to analyze the implicit information in EBSN.

It can be seen from Figure 1 that the Meetup event includes time, location, traffic point, the event object, and event content. Generally, users can filter dislike events based on displayed tags or keywords. Namely, filter out specific information through tags. Simultaneously, there is still some hidden information in the activity. For example, the note part of the event plan lists the nationality ratio of people participating in the event—60% of Japanese locals, and 40% of people from other countries. Although this note may seem a trivial point for a dinner and friendship event, it is possibly of great value in an event that has other language learning needs. This means that not only classroom-type language learning can be recommended, but nonclassroom-type learning scenarios can still be implemented. However, the learning way in this environment will not be found in traditional semantic or keyword-based recommendations. The limitations in the traditional event recommendation invisibly limit people's social choices.

In conclusion, the purpose of the event is to meet different user needs. Given the traditional recommendation algorithm like a content recommendation or collaborative filtering recommendation, it is hard to balance the influence of distinct individual user preferences on implicit information. Therefore, this paper uses IIDQN, a method in reinforcement learning algorithm, to find hidden nodes in user behavior through neural networks. In this way, the implicit information in the user's interest is obtained. As in this example, a Japanese with foreign language learning needs, besides caring about his or her language learning events, will gradually pay attention to language-related activities in his or her browsing trajectory, for example, activities involving foreigners. This attention does not belong

to any interest point in the recommendation history. But it can be seen as an implicit activity recommendation point. Users' social choices will be expanded gradually.

2.2. Sample Noise Problem. The recommended sample data sets are often from a wide range of sources. The EBSN website is generally based on user clicks, browsing time, user feedback, user secondary participation rate, and so on. But for recommendations based on page clicks and browses, there is often a lot of data noise in the data set. For example, mistaken clicks caused by the user's hand sliding, pages with attractive titles or cover pages that attract users, special activities bound pop-ups carried out by website operators, etc. These data are called sample noise. Although the amount of sample noise data is not large, some websites have more sources of sample noise data. A parameter related to browsing time is added in the reward calculation, which excludes click data that has nothing to do with the user's real browsing behavior.

3. IIDQN: Incentive Improved DQN

3.1. Definition of EBSN Networks. The concept of EBSN is first proposed in Ref. [1], which is expressed as a heterogeneous network that includes online and offline relationships $G = \langle U, e_1, e_2 \rangle$, where $U \in U\{u_1, u_2, \dots, u_n\}$: represents a collection of all users.

- (1) $E_1 \in e_{\text{online}}\{e_1, e_2, \dots, e_n\}$, which means the collection of all online users
- (2) $E_2 \in e_{\text{offline}}\{e_1, e_2, \dots, e_n\}$, which means the collection of all offline users

It can be simply regarded as online and offline parts of the networks: $G_1 = \langle U, e_1 \rangle$ $G_2 = \langle U, e_2 \rangle$

Liao et al. divided the framework of the EBSN recommendation system into three layers [1]: data collection layer, data processing layer, and recommendation generation layer. The data collection layer is used to obtain various data information. The data processing layer performs pre-processing operations on the data. The recommendation layer recommends the system according to different recommendation algorithms. Compared with traditional social networks, EBSN has the following characteristics: events and user interests have a heavy-tailed distribution, event participation is heavily dependent on location characteristics, event life cycles are short, missing user display preferences, online networks are more densely distributed than offline networks, and so on. Based on these characteristics, we use IIDQN in reinforcement learning to solve recommended update timeliness and short event declaration period in EBSN.

3.2. Algorithm Calculation Equation. We associate the EBSN recommendation model with the reinforcement learning model. Reinforcement learning defines agent and environment. The agent perceives the environment and rewards

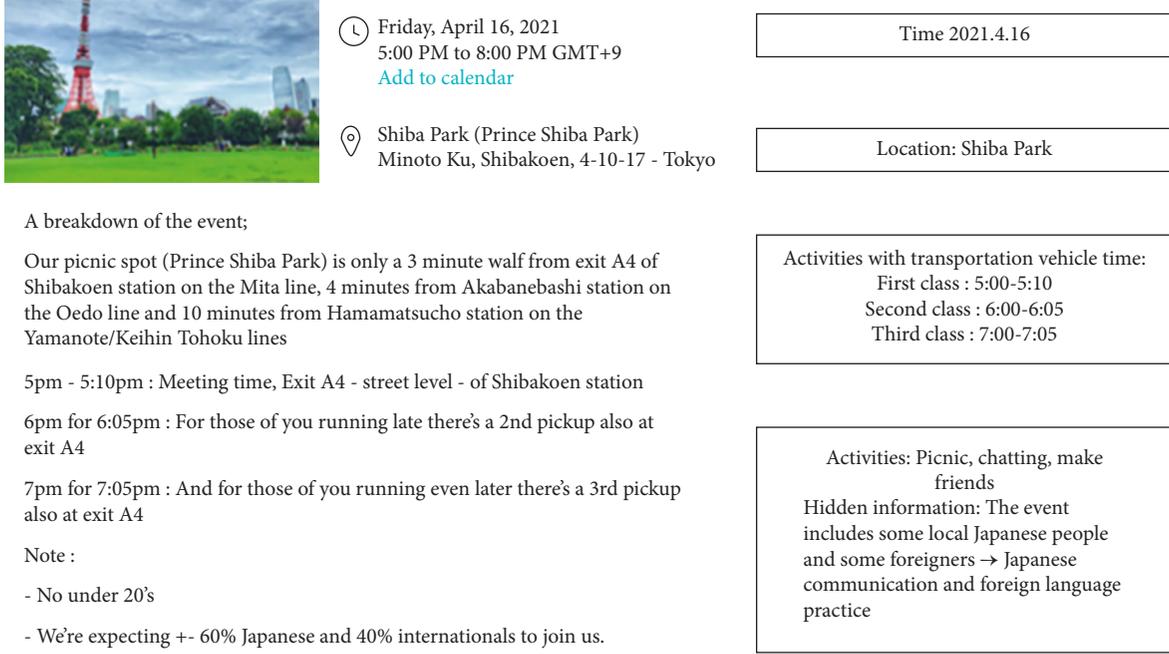


FIGURE 1: Recommended accuracy problem description diagram.

given by strategy changes, learns, and makes the next decision through environmental changes. The ultimate goal of reinforcement learning is to obtain the optimal strategy.

Currently, we define the triples in reinforcement learning. Formally, reinforcement learning consists of a tuple of three elements (S, A, R) as follows:

S is a state space, $S_t \in S$ represents the state in time t , which comes from the user's previous historical information records.

A is an action set, $a_t \in A$ represents the user's recommended choice at time t .

R is a reward matrix, $r(S_{t+1}|S_t, a_t)$ is a direct reward to Agent in state transition probability $P(S_{t+1}|S_t, a_t)$.

In IIDQN, we divide the reward value into two parts: user browsing click rewards and the length of browsing time rewards.

In this system model, we regard the recommendation part in EBSN as an Agent. The interaction process between the recommender system and the person is regarded as the change process of the Agent and the Environment. This interaction process is embodied as a Markov decision process. When the state sequence $S = \{s_0, s_1, s_2, \dots, s_t\}$ satisfies $P(S_{t+1}|S_t) = P(S_{t+1}|S_t, S_{t-1}, \dots, S_0)$, the state at the next moment merely depends on the state at the current moment.

In this model, we use the improved IIDQN algorithm based on DQN that combined Q-learning algorithm and neural network as the algorithm for the agent to accept environmental changes. Q-learning algorithm is temporal-difference learning in reinforcement learning. Temporal-difference learning can solve the model-free sequence decision problem in the Markov decision process. In the recommended situation, it is often difficult to know the state

transition probability of all actions in each state. Therefore, it is a wise choice to use the Q-learning algorithm. Its overall goal is to obtain a reward by simulating a sequence, and to obtain the maximum expected return $V(s)$ in this state by maximizing the reward:

$$V(s) = E\left(\sum_{t=0}^{T-1} \gamma^t r_{t+1} | s_0 = s\right). \quad (1)$$

Equation (1) is called the value equation, where $\gamma \in [0, 1]$ represents the discount rate. When γ approaches 0, it means that the agent is concerned about short-term returns, and when γ approaches 1, it means that the agent is more concerned about long-term returns. Equation (1) reflects that the expected return of the current state can be expressed by the expected return of the next state. Therefore, the maximum reward obtained in the current state is calculated by calculating the reward at the next moment. In addition, in the EBSN recommendation interaction process, due to the known feedback action of the user's recommendation, namely, the action selected in this state in each round of state transition is known. Therefore, we introduce the Q function of the strategy π to consider the action in the current state. The difference between the Q function and the value equation is that the Q function determines the action a in a specific state:

$$Q^\pi(s_t, a_t) = E[r_1 + \gamma Q^\pi(s_{t+1}, a_{t+1}) | s_t, a_t]. \quad (2)$$

The equation (2) is Q value function. Equation (2) determines the action in a state s_t which is related to the future state.

3.3. Markov Modelling. In order to use the reinforcement learning algorithm to address the event recommendation problem in EBSN, the recommendation problem is modeled first. The Markov recommendation conversion between simple events in EBSN is shown in Figure 2. Table 1 explains the corresponding state, action, and reward value.

In Figure 2, S_0 represents the initial state. The recommendation system acts as an agent and the user acts as the environment. In this state, the recommendation system recommends events activities to users. If the user is interested in a certain event and clicks to view the behavior, we will give a certain reward value. On the contrary, if the user ignores the recommendation and browses through search or other categories, it indicates that the user is not interested in the current recommended event. A slight penalty will be given at this time. Therefore, when making a policy selection to the system later, it is very likely that the system will no longer recommend this type of event activity to the user. When a user sees an event that fits the user's interest during the browsing process, and is ready to join the event, the recommendation system to the user is in line with the user's interest. In this state, the reward for the recommendation is great, such as the reward is 10. Therefore, the recommendation system will recommend events with similar characteristics to the event in the next recommendation.

3.4. Redefinition Reward Calculations Based on Time Parameters. In order to distinguish whether the user's browsing behavior comes from their real preferences (that is, the question raised in Problem Description 2), this paper considers the influence of browsing time on the accuracy of recommendation. In addition, the reward function is defined as a linear function related to browsing time. According to different browsing times, the correct recommendation situation is redefined. The reward value will continue to accumulate over time. In most cases, users browse according to their desire to choose their interests. However, it cannot be ruled out that users are affected by the sample space due to title interests, image interests, or other wrong click operations. Therefore, giving a certain reward value to the browsing time can distinguish the error caused by the user's mistaken click operation during the click and browse process. To a certain extent, it solves the reward calculation problem caused by the wrong click operation. Figure 3 shows a schematic diagram of the overall algorithm flow with IIDQN's mutual correspondence between agents, environments, and states.

The reward in the browsing state is linearly added to the user's scrolling browsing time, and r_t represents the additional reward based on the browsing time when the user performs the browsing state. In other words, the total reward for browsing a single event is:

$$r = \alpha * t (r < 6). \quad (3)$$

Among them, α is the reward coefficient, which means that the reward value obtained with the increase of scrolling time increases gradually, and the total r of the additional

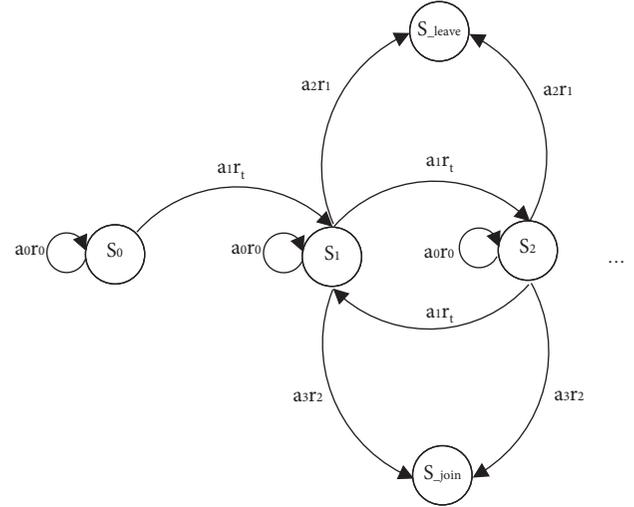


FIGURE 2: Markov state diagram.

TABLE 1: Interpretation table of the Markov state diagram.

Status set	
S_0	Initial status
S_1	Browse status
S_2	Browse status
...	...
S_{leave}	Leave website status
S_{join}	Add activity status
Action set	
a_0	No browsing
a_1	Browse recommendations
a_2	Leave the website
a_3	Join the activity
Reward set	
r_0	0
r_1	F02D 10
r_2	10
r_t	$\lambda * t$

reward value and the original reward cannot exceed a certain window value. Assuming the window value is set to 6 in the initial state, the purpose of setting the window value is to make the upper limit of the browsing time reward not exceed the reward obtained by joining the event.

Table 2 represents a simple reward calculation process:

Assuming four recommended events under this recommendation model, four simple events $e_1, e_2, e_3,$ and e_4 are recommended in the initialization state. Environment perception and selection action mean that the user chooses to browse e_2, e_4 according to his or her interests and other attributes finally join event e_2 . In this event, the state S represents the recommendation of the four events, and the environmental action A is that the user is in the state S_0 . The user makes the actions of browsing e_2, e_4 as the environment and transitions to the state S_1 . The reward obtained is the reward $R = \alpha * t_{e_2} + \alpha * t_{e_4}$ obtained according to the browsing process of e_2 and e_4 .

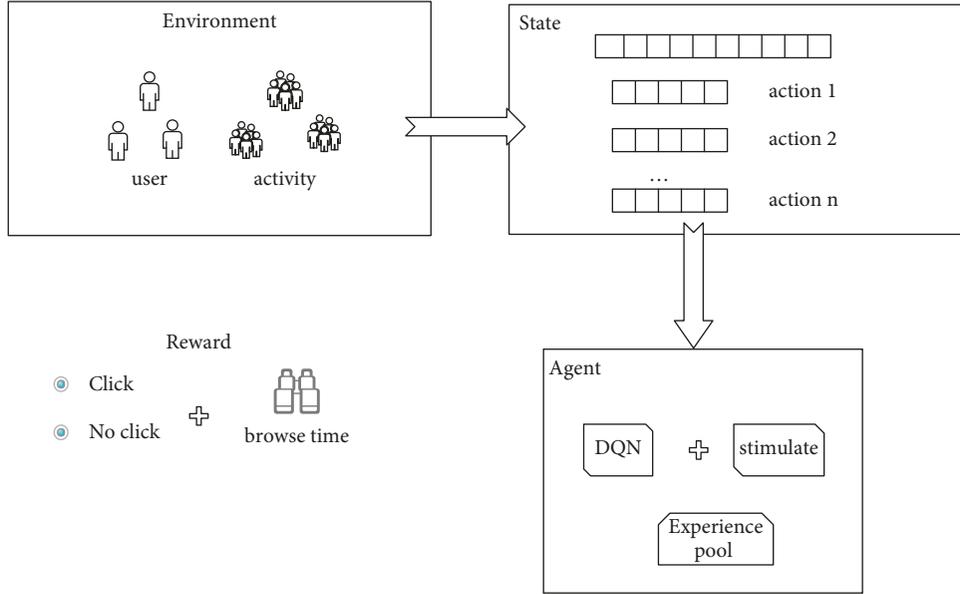


FIGURE 3: Schematic diagram of algorithm model.

TABLE 2: Event example.

	$a = \text{browsing}$	$a = \text{join event}$
e_1	-1	0
e_2	$r = \alpha * t$	+10
e_3	-1	0
e_4	$\alpha * t$	0

3.5. Reward Calculation in the Case of Feature Sparse. In order to solve the problem of the sparse number of sample features in the sample, this paper redefines the reward function in Q-network, hoping to mine the hidden information in the event to recommend the user (the impact on problem description 1). This article analyzes the initial situation when the hidden features appear, offering additional rewards under different conditions where the hidden features just appear. Therefore, weight value is constantly changing during the weight update process of the neural networks, which represents the influence of a certain feature on the final recommendation result. Specifically, we store the weight of each iteration in a matrix to calculate the weight change rate of the K iteration processes. If the rate of change rises rapidly, indicating that it is a hidden feature, a slight reward is given according to equation (4). Contrarily, if the rate of change decreases slowly, it indicates that it is not a hidden feature but may be an error value. We give a slight penalty for this change. The newly appearing networks node with a smaller weight is used as a hidden feature for additional rewards. Suppose the minimum weight characteristic value is β , and the normal sample characteristic value B has:

$$R(s_t, a_t) = \begin{cases} \frac{\alpha}{b}, & s \in D, \\ -\frac{\alpha}{b}, & s \in D', \end{cases} \quad (4)$$

where D represents the correctly classified sample set, and D' represents the incorrectly classified sample set. When the b of a certain type of sample is close to the minimum sample characteristic value β and much smaller than the normal characteristic value B :

$$B \gg b \geq \beta + \varepsilon (\varepsilon \text{ is a very small number}). \quad (5)$$

We use equation (3) to calculate the reward. Let $1/b \in (0, 1]$, where the sparse reward value increases with the sparseness of the feature value. When the sparse critical value is reached, the reward obtained is infinitely close to α .

The abscissa represents the sparseness of the interest feature, and the ordinate represents the reward value obtained in the range of the feature.

3.6. Overall Calculation Flow of IIDQN Algorithm. This section describes the algorithm process of the deep Q network using reinforcement learning based on incentive improvement in the recommendation system.

Figure 4 shows the specific process implementation scheme of IIDQN. Next, we discuss the specific algorithm implementation. Figure 4 shows the overall calculation process of the IIDQN algorithm.

In this model, in the initialization state, we define an experience pool and two networks with the same structure. [lines 1,2] One is called the Q network for each round of model iteration calculations, and the other network is called the target network. The parameter value in the target network is used as the final calculated true value. Select the recommended item a [lines 3,4] in the initialization state s . Next, proceed to the part of the recommendation system interacting with the user [line 5]. The reward function of this action will take into account the time incentives mentioned in Section 3.4. Then, put the quadruple (S, A, R, S') into the experience pool. Sample the experience in the experience

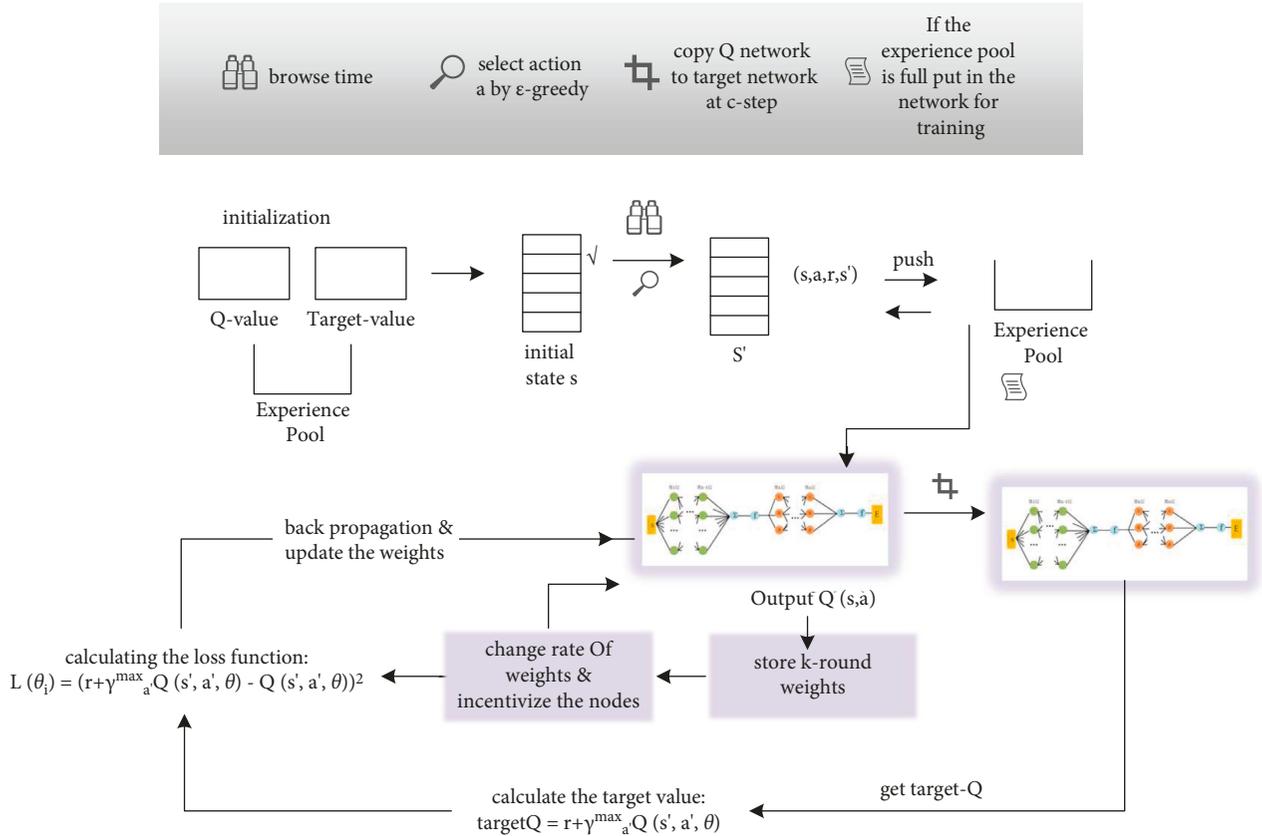


FIGURE 4: The calculation process of IIDQN in EBSN.

pool through experience replay [lines 6–7]. During Q network training, N pieces of experience are randomly selected from the experience pool and placed in the network for experience selection. That is, the network input of the Q network is in N states s. The final output value obtained by the Q network is the expected reward Q value of all actions that can be selected in each state. Accordingly, the Q-value is calculated through the neural network, and the Q network is recalculated using the square loss function [line 8]. Identify sparse nodes and incentivize the sparse weights in the range [line 9]. Redefine a few classified samples in the data set by storing feature values with smaller weights in nodes. The change rate is dynamically calculated for each round of the weight characteristic value, and the nodes with smaller weight values are stimulated. In this way, implicit information is recognized. In order to maintain the stability of the calculation results in the network, the parameters in the Q network are copied to the target network for storage processing every c step [line 10]. The calculation of the target value is performed according to the stored Q value in the target network, and the calculation process is the calculation process of the true value in the Q-learning algorithm. The specific algorithm flow is shown in Algorithm 1, where the bold part is the difference between the IIDQN and DQN algorithms:

4. Experiment Results and Analysis

4.1. Experiment Data Description. The recommendation problem of a single user in the event network is analyzed. The data of this experiment are from the data set of the meetup website. The recommendation changes with the user’s preferences, and finally mean reward is obtained.

This data set classifies the 36 types of activity group interests in a meetup in detail and divides the browsing time of each user’s activity browsing event in detail.

Behavioral strategy: the experiment’s DQN algorithm strategy based on reward transformation adopts the epsilon-greedy exploration strategy.

Next, we discuss several important related parameters. For example, in Section 3 we introduce parameter α . The reward factor can choose to be 0.05, that is, if the user’s browsing time is 120 s, the reward value is 6. We use the user’s browsing time of 120 s as the limit, and set the reward threshold to 6. Over 120 s, we default that the user is interested in the current browsing, and no additional bonus value will be added. In addition, we use the ϵ -greedy strategy to explore the information in the experience pool, by doing so to ensure the recommendation results are independent and identically distributed. The initial exploration ϵ is 0.6, and the coefficient will continue to decrease as the agent

IIDQN algorithm

Inputs: state space S , action space A , discount rate γ , learning rate α , parameter update interval C .

- (1) Randomly initialize the parameters θ of the Q-networks and randomly initialize the parameters θ of the target Q-networks.
- (2) Initialization state S .
- (3) Select action a .

(4) **Perform action, get reward r and next action S_0 through environment.**

(5) Put S, a, r, S_0 into the experience pool and sample

(6) Sample ss, aa, rr, ss_0 in the experience pool for the next action. Let y be the target value

$$y = \begin{cases} rr, & \text{ss' is the termination state} \\ rr + \gamma \max_{a'} Q(ss', a'), & \text{or} \end{cases}$$

(7) Update the weights by back propagation mechanism and retrain the Q networks using gradient descent algorithm

(8) **Detect sparse nodes and apply additional reward/penalty updates to nodes**

$$\mathbf{R}(s_t, a_t) = \begin{cases} \mathbf{a}/\mathbf{b}, & s \in D \\ -\mathbf{a}/\mathbf{b}, & s \in D' \end{cases}$$

(9) Update the target networks every N steps and copy the current networks parameters to the target networks.

ALGORITHM 1: Incentive improvement algorithm based on Q networks in EBSN.

continues to learn, and the termination exploration ϵ is 0.05. This shows that more attention is paid to the exploration of newly added data in the initial stage, so the size of the DQN experience pool cannot be too small.

4.2. Comparison of Recommended Models. To find the most suitable recommendation algorithm under the EBSN model, we compared several proposed frameworks, including the original DQN algorithm. To evaluate the recommendation performance, we divide each data set into a training set and a test set, use 80% of them as the model training set, and train 20% of the model data.

CF: The collaborative filtering recommendation method mainly benefits users by displaying user information on different preferences and predicts information by finding users with similar preferences.

DNN: Deep Neural Networks, which use neural networks to predict user preferences, are also the user's historical data recommendation information, and the output is the DNN output recommendation item.

RNN: It is a type of sequence data input, recursive in the evolution direction of the sequence, and all nodes (cyclic units) are connected in a chain.

DQN: We first use the DQN algorithm for recommendation prediction, determine the correspondence between the agent and the environment, and input the user's historical information as the state.

Improved DQN: The improved DQN algorithm for incentives is proposed in Section 3 of this article.

The above types of recommendation models have a wide range of choices. There are traditional recommendation models and deep learning framework recommendations, and model recommendations in reinforcement learning. The DQN algorithm is chosen as the baseline because the DQN algorithm can continuously update the strategy during the interaction process. DNN and RNN are cited as a comparison based on the contrast gap between neural networks in the DQN algorithm. RNN can capture the time series of

the user's browsing history because the order of recommended browsing on the product page can affect each other. Therefore, we introduce RNN to consider this reason.

According to the effect of the above several recommended models on the simulator, we use NDGC [25] and MAP [26] as the two evaluation criteria for comparison. NDCG is a normalized DCG, which is an evaluation index for measuring search recommendations. This indicator takes into account the relevance of all elements. MAP (mean Average Precision) is an indicator of recommended accuracy. It is calculated by summing the mean accuracy of all categories and dividing by all categories. Figure 5.

According to the recommended data in the above figure, it can be seen that:

- (1) Generally, the recommendation efficiency of using deep learning and reinforcement learning frameworks is significantly better than the general recommendation results. To a certain extent, the traditional recommendation model represented by CF ignores the time interaction factor in the user input information. Since traditional models pay more attention to user characteristics, they are not suitable for interaction-based recommendations in EBSN.
- (2) In addition, comparing the deep recommendation model (DNN, RNN) and the reinforcement learning recommendation model (DQN), we can find that reinforcement learning still performs better than the deep learning recommendation to a certain extent. Because deep learning pays more attention to recommending activities that can increase the model's timely rewards, the model promotion in reinforcement learning will merge the user's rewards throughout the participation cycle. The DQN model focuses on the overall user experience from the beginning of user registration to a long time in the future. The internal user's overall experience will be quantified as the total revenue of the model.

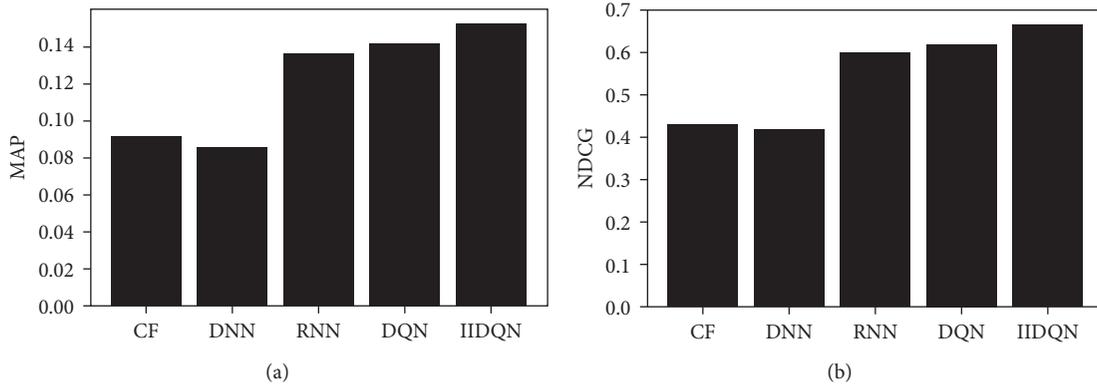


FIGURE 5: Performance comparison in different recommendation systems. (a) Overall performance comparison on MAP. (b) Overall performance comparison on NDCG.

- (3) Based on the comparison of the above recommendations, we can see that the model selected for reinforcement learning is better in our recommendation to solve this problem. Nevertheless, only using DQN cannot solve some of the problems mentioned in the previous problem description. Therefore, we finally improved the DQN algorithm and finally got a recommendation effect better than DQN.

5. Blockchain-Based Activity Methods

As the core of the blockchain, the consensus algorithm ensures the mutual trust relationship between nodes in the blockchain and thus maintains the security of the blockchain. However, offline users in the event network are mostly strangers, and it is difficult to establish a trust relationship between them. Therefore, using the characteristics of blockchain technology can guarantee the mutual trust relationship between nodes, and we consider the new problem of recommendation in EBSN: The recommended description of online activities is inconsistent with the actual offline activities. In order to solve this problem, in this section, we propose a deposit consensus mechanism based on blockchain technology. The problem of whether the activities in the EBSN network conform to the activity recommendation is modeled on the blockchain system, and the deposit consensus mechanism is used to solve the problem.

5.1. Model Overview. Based on the scattered and complex characteristics of event network nodes, blockchain technology is applied to the event network, and the entire network is regarded as a scattered blockchain node. The behavior information generated by all users in the event network will be written on the chain for recording.

There are two kinds of membership in the network: sponsor and participant. Correspondingly expressed as two kinds of user nodes on the blockchain, sponsor is the event initiator of each activity. Sponsor needs to obtain the consent of a few validators before proceeding with the actual event initiation. Validator is a few randomly generated validators in

the chain that are used to verify the identity of the sponsor and vote whether the activity proposed by the sponsor is on the chain. These few randomly generated nodes are equivalent to the identities of temporary supervisors, ensuring the fairness and security of activities among nodes in the entire network. When the sponsor creates an activity, a new consortium chain is generated. The address of the consortium chain and the users' name that caused him or her to be generated will be recorded on the public chain. Each block on the alliance chain records an event activity information, which includes the trust deposit of the organizer, the overall process recorded in the activity event, and the activity transaction fee submitted by the user. Figure 6 shows the main activity function of event activity group A in the EBSN blockchain network.

For other user participants in the chain, when participants want to participate in an activity, they will apply to the legal activity sponsor to join the group in the same way. After the validator in the activity group agrees to join the group, all activity transactions and activities during the activity will be written on the alliance chain within the organization.

5.2. Build Model. We construct the current activity relationship in the event network as a network relationship in the blockchain. In addition, the set $E = \langle b, U, A \rangle$ in the event network. Among them: b represents the block number, $U \in \{U_s, U_p\}$ represents the set of all users and divides the set of all users in the network into two categories, event sponsor (u_s) and event participant (u_p). $A \in \{a_1, a_2, \dots, a_n\}$ represents the set of all alliance chains.

In the network, for the number of z users, user u corresponds to z nodes in the blockchain network. And, there is a public chain and multiple alliance chains in the network. The public chain records the information of all event groups, and the alliance chain records the process information of each event group in the entire event activity. It mainly contains information such as time of initiation, specific details of event activities, transaction records of activity fees for members to participate in the event, and credit deposit submitted by the event initiator, etc.

For sponsor users u_s who want to create an activity in the blockchain, they first need to publish an event activity

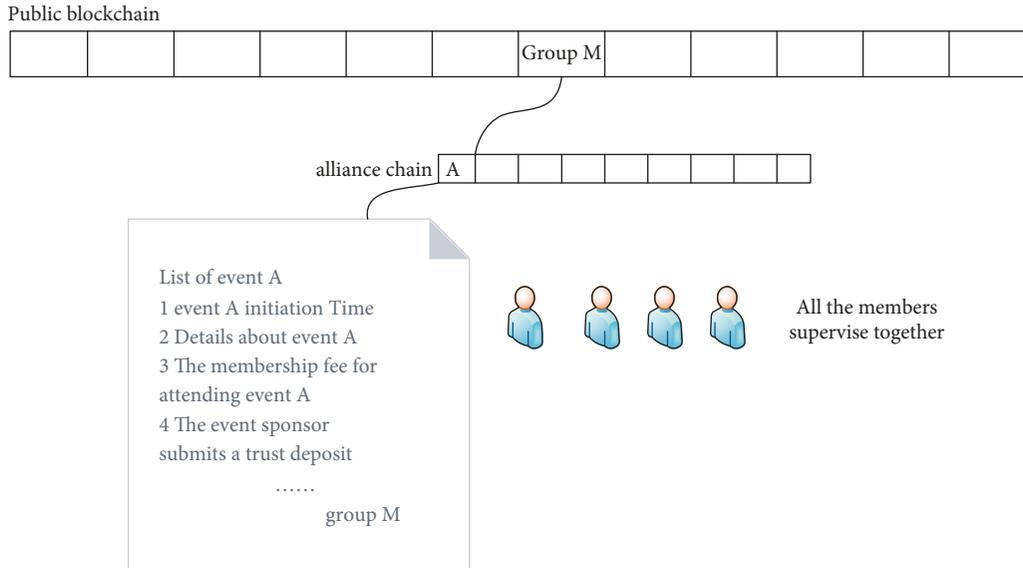


FIGURE 6: Examples of blockchain-based activities.

application information to the public chain. In addition to the user's own personal identity verification information and event activity details, the application information also contains how much money the user decides to spend as the trust deposit for this activity. If more than half of the verification nodes pass the verification and agree to the user's activity creation, then $u_s a$ will successfully create the activity event M , and will generate a consortium chain corresponding to the activity event M . On the public chain, the user information of the initiator of the event, the corresponding consortium chain address, and trust deposit will be recorded. In the alliance, it is mainly used to record the actual information experienced by the event activity offline. In this way, it not only guarantees the safe conduct of offline activities but also ensures the consistency of online event network recommendation descriptions and offline event activities.

5.3. Trust Deposit. Leverage the immutability feature of the blockchain to ensure that the online recommendations seen by participants are consistent with offline activities. For other user participants present in the chain, the event organizer will take a part of the amount as a trust deposit when creating the event, and put the trust deposit on the blockchain. After the event, each event participant will rate and score the entire event process. At the same time, the most important thing is to score whether the activity meets the description of the activity initiator on the web page. This will be used as a very important indicator for the recommendation system to make recommendations, to promote the evaluation of the overall activity experience by online recommendations. Finally, the average value of this indicator is used to quantify the event organizer. Here, we take into account the influence of different people on the evaluation preferences, and use the overall variance to process the calculation. According to the user score, it is finally determined how much the trust deposit originally placed on the chain by the initiator of the event can be recovered. If the

score is not satisfactory, the event organization is very likely to be seriously inconsistent with the original description, and then the event organizer will receive negative feedback from the participants. In addition, the recommendability of the event organizer will be weakened, and the original trust deposit on the chain will also be deducted.

In order to prevent malicious evaluation by event participants, the overall evaluation process follows the consensus protocol PBFT in the blockchain. Practical Byzantine fault tolerance (PBFT) is one of the consensus algorithms proposed earlier [27]. As a practical consensus algorithm based on state machine, PBFT's role model can correspond to the organizers and participants in the event network. Although the consensus mechanism can ignore the malicious influence of a certain user to a certain extent, it is also difficult for the algorithm to achieve consensus if malicious commenters exceed one-third of the total participants.

6. Conclusions

EBSN is a field of promising research today, which is of great significance from online and offline security research. This paper offers a mechanism based on blockchain technology in EBSN, which includes creating activities on the organizer chain and recommending online and offline event activities guaranteed by blockchain. Simultaneously, offline activities conform to the online recommendation description through the blockchain. Furthermore, we add the reinforcement learning algorithm to the event recommendation, improve the DQN algorithm, and propose IIDQN. Through this algorithm, the recommendation process of dynamic interaction can be simulated. Improve the time-related parameters to eliminate sample noise in the recall phase. However, the work done in this paper needs further research. For example, it is only compared with a few typical recommendation algorithms, and the differences between other algorithms are not considered. In addition, we merely considered the impact of time in this

study, and there are many other factors that will affect the final recommendation accuracy. For the overall algorithm, it has only been verified in a small scale. In the following work, it should be further extended to a more general situation for further analysis and research.

Data Availability

The CSV data used to support the findings of this study are available in <https://www.kaggle.com/stkbailey/nashville-meetup>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This study was supported by the Foundation of National Natural Science Foundation of China (Grant Number: 62072273, 72111530206, 61962009, 61873117, 61832012, 61771231, and 61771289), Natural Science Foundation of Shandong Province (ZR2019MF062), Shandong University Science and Technology Program Project (J18A326), Guangxi Key Laboratory of Cryptography and Information Security (No: GCIS202112), The Major Basic Research Project of Natural Science Foundation of Shandong Province of China (ZR2018ZC0438); Major Scientific and Technological Special Project of Guizhou Province (20183001), Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2019BD-KFJJ009), and Talent project of Guizhou Big Data Academy. Guizhou Provincial Key Laboratory of Public Big Data ([2018]01).

References

- [1] G. Q. Liao, T. M. Lan, and X. M. Huang, "Survey on recommendation systems in event-based social networks," *Ruan Jian Xue Bao/Journal of Software*, vol. 32, no. 2, pp. 424–444, 2021.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning," *Computer Science*, <https://arxiv.org/abs/1312.5602>, 2013.
- [3] V. Minh, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
- [4] C. J. C. H. Watkins, "Learning from delayed rewards," *Robotics and Autonomous Systems*, vol. 15, no. 4, pp. 233–235, 1995.
- [5] X. Wu, Y. Dong, B. Shi, A. Swami, and N. V. Chawla, "Who will attend this event TOGETHER? Event attendance prediction via deep LSTM networks," in *Proceedings of the 18th SIAM International Conference on Data Mining (ICDM)*, pp. 180–188, University of Illinois, Chicago, IL, USA, 2018.
- [6] A. Van Den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," *Advances in Neural Information Processing Systems*, vol. 26, no. 2, pp. 2643–2651, 2013.
- [7] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck, *Temporal Pooling and Multiscale Learning for Automatic Annotation and Ranking of Music Audio*, Ismir, Canada, 2011.
- [8] Y. Zuo, J. Zeng, M. Gong, and L. Jiao, "Tag-aware recommender systems based on deep neural networks," *Neurocomputing*, vol. 204, pp. 51–60, 2016.
- [9] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *Computer Science*, abs:1511.06939, 2015.
- [10] W. Caihua, J. Wang, J. Liu, and W. Liu, "Recurrent neural network based recommendation for time heterogeneous feedback," *Knowledge-Based Systems*, vol. 109, no. 1, pp. 90–103, 2016.
- [11] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," *Journal of Machine Learning Research*, vol. 15, pp. 29–37, 2011.
- [12] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the International Conference on Machine Learning ACM*, pp. 791–798, Corvallis Oregon, June 2007.
- [13] Y. Wang and J. Tang, "Event2Vec: learning event representations using spatial-temporal information for recommendation," in *Proceedings of the 23rd Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, pp. 314–326, Macau, China, April 2019.
- [14] Z. Wang, Y. Zhang, H. Chen, Z. Li, and F. Xia, "Deep user modeling for content-based event recommendation in event-based social networks," in *Proceedings of the 2018-IEEE Conference on Computer Communications*, pp. 1304–1312, 2018.
- [15] L. Luceri, T. Braun, and S. Giordano, "Social influence (deep) learning for human behaviour prediction," in *Proceedings of the International Workshop on Complex Networks*, pp. 261–269, Honolulu, HI, USA, April 2018.
- [16] S. Y. Chen, Y. Yu, Q. Da, J. Tan, H. K. Huang, and H. H. Tang, "Stabilizing reinforcement learning in dynamic environment with application to online recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1187–1196, London, UK, July 2018.
- [17] G. Zheng, F. Zhang, Z. Zheng et al., "DRN: a deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176, Lyon, France, April 2018.
- [18] X. Zhao, L. Zhang, and Z. Ding, "Recommendations with negative feedback via pairwise deep reinforcement learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1040–1048, New York, NY, USA, 2018.
- [19] J. She, Y. Tong, C. Lei, and C. Chen, "Conflict-aware Event-Participant Arrangement," in *Proceedings of the IEEE International Conference on Data Engineering IEEE Computer Society*, pp. 735–746, 2015.
- [20] C. Ge, W. Susilo, and J. Baek, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable and Secure Computing*, no. 1, p. 1, 2021.
- [21] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy Re-encryption scheme for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, 2021.
- [22] C. Zhao, S. Zhao, M. Zhao et al., "Secure multi-party computation: theory, practice and applications," *Information Sciences*, vol. 476, pp. 357–372, 2019.
- [23] Y. Lei, S. Chen, L. Fan, F. Song, and Y. Liu, "Advanced evasion attacks and mitigations on practical ML-based phishing website classifiers," 2020, <https://arxiv.org/abs/2004.06954>. 06954.

- [24] T. Li, Y. Chen, and Y. Wang, “Rational protocols and attacks in blockchain system,” *Security and Communication Networks*, vol. 2020, Article ID 8839047, 2020.
- [25] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.
- [26] A. Turpin and F. Scholer, “User performance versus precision measures for simple search tasks,” in *Proceedings of the 29th Annual International ACM SI*, Seattle, WA, USA, August 2006.
- [27] M. Castro and B. Liskov, “Practical byzantine fault tolerance and proactive recovery,” *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.