International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 780-792

https://doi.org/10.70517/ijhsa46358

Research on the optimization method of balancing diversity and accuracy based on Monte Carlo tree search algorithm in e-commerce platform recommendation system

Shuxin Wei[1,*]
[1] Guangdong University of Science and Technology, Dongguan, Guangdong, 523083, China
Corresponding authors: (e-mail: weishuxin@gdust.edu.cn).

Abstract In recent years, with the rapid development of the Internet, the e-commerce platform has become an indispensable part of people's daily life, which can generate personalized recommendations for users to meet their daily life purchase needs. The research adopts the MCTS method combining deep learning and Monte Carlo tree search algorithm, fuses the strategy network, value network and risk network with the strategy value risk network, and combines the improved MCTS method to construct a self-learning strategy value risk network applied to the e-commerce platform recommendation system. The improved MCTS method is verified through experiments, and the improved MCTS method has high comprehensive performance, compares the recommendation effect of the self-learning strategy value risk network algorithm with the UCT algorithm, and utilizes the hybrid reward function to achieve the balance between the accuracy and diversity of recommended products. The improved MCTS method is tested in terms of coverage, diversity and user satisfaction, and the performance is good, with the highest coverage of 0.91 and the highest diversity of 0.9941. In this paper, the system realizes the diversified recommendation of products to meet the user's need for recommender systems.

Index Terms deep learning, recommender system, Monte Carlo tree search, strategy value risk model

I. Introduction

With the rapid development of the Internet, e-commerce platforms have become the main way for people to shop [1], [2]. In order to improve user experience and promote transactions, recommender systems are widely used in various e-commerce platforms to provide personalized recommendation services for users [3]-[5]. Commodity recommender systems are used to recommend commodities that may be of interest to users based on their historical behaviors, personal preferences, and other factors through machine learning, data mining, and other technologies [6], [7]. This kind of recommendation not only facilitates users, but also can increase the transaction volume and stickiness of the platform [8]. However, recommender systems also have a problem of balancing the diversity and accuracy of recommendation results [9], [10].

Recommender systems only recommend items that users already like or are very similar to, the recommended results will lack novelty and diversity, leading to user boredom and missing out on potential consumption of goods, so recommender systems should try to recommend different types of items that users may be interested in while providing relevant items [11]-[14]. However, the opposite of diversity is accuracy [15]. If the recommender system only focuses on diversity and ignores user preferences, the recommendation results do not meet the user's expectations, leading to user distrust of the recommender system; therefore, the recommender system needs to ensure that the recommendation results are accurate and in line with the user's preferences while maintaining diversity [16]-[19].

In order to balance the diversity and accuracy of recommendation results, researchers have proposed some algorithms and methods, and the common method is based on Monte Carlo tree search algorithm [20], [21]. Monte Carlo tree search algorithm is a very practical algorithm, which is widely used in board games, automatic driving, robotics and so on [22], [23]. In the optimization of the balance between diversity and accuracy in the recommendation system of e-commerce platforms, Monte Carlo tree search algorithm achieves the balance optimization of the two by simulating the stochastic nature of the balance between the diversity and accuracy of recommendations [24]-[26].

Literature [27] introduced the current research status of recommender systems and proposed a user-based collaborative filtering based on kernel methods and multi-objective optimization, verified its effectiveness, and improved the accuracy and diversity of the recommender system by improving the concept drift in dynamic data. Literature [28] proposed a structural balance theory based e-commerce recommendation method (SBT-Rec), based

on the product items purchased and favored by the target user, determining their needs based on the structural balance theory and recommending them to the target user, and verified the feasibility of SBT-Rec. Literature [29] analyzed the factors that determine customer satisfaction when using recommender systems, developed several recommender systems and measured their ability to provide accurate and diverse recommendations, emphasizing that the accuracy and diversity of deep learning recommender systems have a positive impact on customer satisfaction. Literature [30] emphasizes the importance of e-commerce, analyzes the value of recommender systems, collaborative recommender systems and their related technologies, and suggests recommendations to customers based on their interests and promotes the ease of search for users to choose the right product for them. Literature [31] introduced a multi-domain item reordering system based on the best interpretation of items based on the shortcomings of collaborative filtering recommendation algorithms, which is an optimal ordering path extracted from the knowledge graph of linked open data connecting the recommended and interacted items, which effectively improves the diversity and accuracy of recommendations. Literature [32] proposed an adaptive trust-aware recommendation model based on users and items, which was revealed through experiments on different datasets to be able to balance and adapt the accuracy of individual and aggregate diversity, and significantly improve the accuracy for cold-start users and long-tail items. Literature [33] constructed a single optimization model based on a matrix completion framework and leveraged available ratings and item metadata to achieve a balance between accuracy and diversity, which achieved higher diversity for a given decrease in accuracy compared to existing techniques. Literature [34] highlights the importance of diversity and accuracy for recommender systems and proposes a two-stage collaborative filtering optimization mechanism to obtain a complete and diverse list of items, and experimental evaluation reveals that the proposed model achieves higher diversity compared to traditional approaches. Literature [35] proposes a new heterogeneous graph neural network framework for diverse recommendation matching to improve the accuracy and diversity of recommendations, which is verified to have accuracy and diversity metrics by launching an extensive evaluation of the recommender system, achieving significant improvements. Literature [36] states that the quality of recommender systems is measured by the accuracy of predictions, and discusses the dynamic effects of diversity, novelty and serendipity of recommended goods, and user's production and consumption behaviors on user stickiness, showing that novelty and serendipity affect user stickiness positively, and more diverse recommendations are detrimental to user stickiness. Literature [37] points out the imbalance between accuracy and diversity of traditional recommendation methods, proposes DivMTID, and improves the effectiveness of recommendation by realizing accurate and diverse recommendations, revealing DivMTID as an effective, accurate, and diverse recommendation method. Literature [38] emphasized the importance of recommender systems for e-commerce platforms, and based on the imbalance between the traditional recommender systems in terms of recommendation accuracy and diversity, proposed the Singular Value Decomposition Multi-Objective Immunity Algorithm, which verified the superior performance of the method to change e-commerce recommender systems, and improve the user experience and the performance of the platform. All the above researches emphasize the importance of diversity and accuracy of recommender systems and propose methods such as SBT-Rec, DivMTID, etc., and the same experiments reveal the effectiveness of the proposed methods, which not only realize the balance between diversity and accuracy of recommender systems, but also effectively improve the user experience.

This paper proposes an improved Monte Carlo tree search algorithm, which combines deep learning with Monte Carlo tree search algorithm, and constructs a self-learning strategy value risk network algorithm by combining the improved Monte Carlo tree search algorithm with the strategy value risk network. Taking the recommendation system of the e-commerce platform as the research object, based on user interest extraction, combined with the self-learning strategy value risk network algorithm is used for platform recommendation. At the same time, a special Top-N list generation algorithm and a hybrid reward function are designed to realize the diversified list-level recommendation of commodities, and the coverage rate, diversity and user satisfaction are tested to verify the effectiveness of the algorithm.

II. Method

II. A.Monte Carlo tree search based on UCB

II. A. 1) Monte Carlo Tree Search Algorithm

Monte Carlo method, also known as computerized stochastic simulation method, is a very important method of numerical computation based on probabilistic statistical theory for know. The basic idea is that the results of a large number of repeated trials or sampling can be used to replace a random variable when it is being counted [39].

A Monte Carlo tree search implementation based on the UCT algorithm is constructed. This algorithm works so well in games with high branching factors that it quickly became the main method for Go games. The algorithm

works well in games with large branching factors and can be stopped at any time, and the longer the time the better the results, providing a high degree of flexibility.

The basic Monte Carlo tree search algorithm is very simple: the search tree is constructed according to the nodes based on the output of the simulation. The simulation process of the search tree:

1) Selection: Starting from the root node, the node selection algorithm recursively selects the optimal child nodes until it reaches a node that is most worthy of expansion, i.e., a node that represents a non-terminal state of the game and has not yet been expanded.

2) Expansion: if the node is not a terminating node (i.e., does not lead to the termination of the game) then one or more child nodes are created based on the currently available actions, one of which is chosen.

3) Simulation: run a simulated output from this node based on the default strategy until the game ends.

4) Backpropagation: use the results of the simulation to update the current node as well as its parent recursively.

The simulation process of the Monte Carlo tree search algorithm can be summarized in two different strategies:

(1) Tree strategy: select or create a leaf node from the search tree (selection and expansion phases).

(2) Fast walk sub-strategy: from a non-terminal node according to which the strategy is played until a result, i.e., a payoff value, is obtained (simulation phase).

II. A. 2) UCT algorithm

The upper bound confidence algorithm defines a UCB value for each rocker arm in the K-arm bandit machine, and replaces the UCB value with the original expected return value each time as the basis for selection. The rocker arm selection algorithm is specifically described in equation (1).

$$UCB_i = \overline{X_i} + c\sqrt{\frac{\ln N}{T_i}} \tag{1}$$

where $\bar{X}_i$ represents the expected return (the average of historical returns) of each rocker arm in the history of the rocker arm, and this value represents the "utilization" in the "exploration and utilization" problem, that is, the use of historical information; $c\sqrt{\frac{\ln N}{T_i}}$ represents the confidence interval of the expected return in the previous part, that is, the "exploration" in the "exploration and utilization" problem, which indicates the robber's confidence in the existence of a better solution for the rocker, and if the confidence interval of the rocker arm is larger, the more likely there is a better solution and the more willing to choose the rocker. This algorithm can make the robbers more willing to find a better solution to achieve the goal of finding the global optimal solution.

The UCT algorithm is the UCB algorithm used in the selection phase of the game tree search algorithm, i.e., the Monte Carlo tree search algorithm can be used as an example to express the UCT algorithm simply in the form of the formula UCT=MCTS+UCB. In the selection phase of the game tree search algorithm, each node in the game tree is regarded as a bandit, and the branch to be selected is the rocker arm, and the selection phase recursively examines each child node to Select an optimal node, the criterion for examination is the UCB algorithm, i.e., calculate their UCB values for each child node and select a child node with the highest UCB value. In practice, the UCB algorithm has been combined with the Monte Carlo tree search algorithm to solve the “exploration and utilization” problem in the Monte Carlo tree search algorithm.

II. B.Monte Carlo Tree Search Based on Deep Learning

The Monte Carlo tree search algorithm is combined with the policy obtained from deep learning in the process shown in Fig. 1. In the new Monte Carlo tree search algorithm, each search edge (s,a) stores the action value $Q(s,a)$, the number of accesses $N(s,a)$, and the a priori probability $p(s,a)$, where the action value $Q(s,a)$ is the current valuation of the branch (s,a) after the Monte Carlo tree search algorithm, and the a priori probability $p(s,a)$ is the estimation of the current board disk by the deep learning obtained from the supervised strategy π for the estimation result of the current chessboard disk. As shown in Fig. 1, each simulation of the Monte Carlo tree search algorithm starts from the root node, and at t steps of the simulation, the action a_i is selected according to Equation (2).

$$a_t = \arg\max_a (Q(s_t,a) + u(s_t,a)) \tag{2}$$

where the reward value $u(s,a)$ and the action value $Q(s,a)$ are as follows according to the UCT algorithm equation (1):

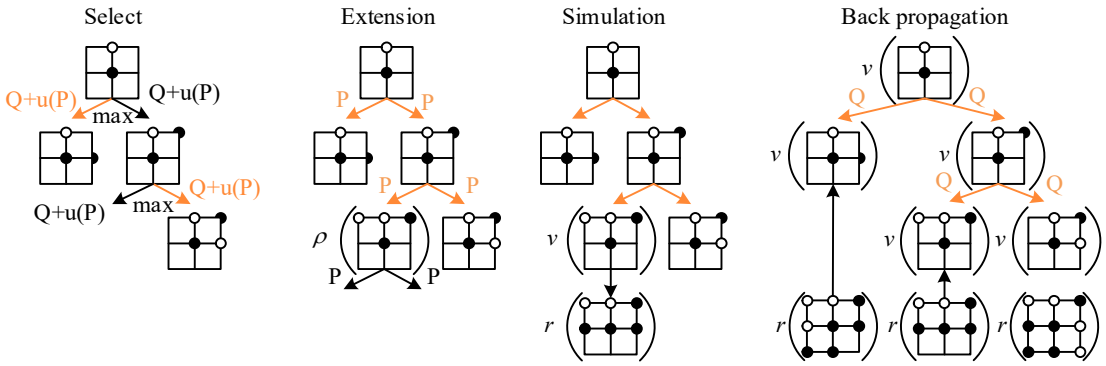


Figure 1: The monte carlo search is combined with the deep learning strategy

$$u(s,a) = c \cdot p(s,a) \cdot \frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)} \tag{3}$$

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{n} 1(s,a,i) \tag{4}$$

From the above equation, $u(s,a)$ is proportional to the strategy network result $p(s,a)$, but the value of this component decreases after multiple visits to the branch (s,a), which encourages the algorithm to explore other less visited branches. When the Monte Carlo tree search simulation reaches a leaf node, the leaf node is first expanded according to the policy network; after expansion, each new leaf node is simulated for valuation, and the result z_L is obtained by fast simulation according to the chosen fast walk sub-method; the results of the simulation are backtracked in the fourth stage of the Monte Carlo tree search algorithm to update the action value and the number of visits for each branch (s,a) of the action values with the number of visits.

II. C.E-commerce platform recommendation strategy value risk network design

II. C. 1) Strategy Network

Reinforcement learning is an important branch of machine learning which focuses on solving continuous decision making problems. The strategy network provides the recommendation probability of each behavior. The first step to design a policy network is to build a neural network model that directly outputs recommended values for predicting individual behaviors by inputting the observed state of the environment.

The structure of a simple strategy network is roughly, 3 layers of fully convolutional neural network using 32, 64 and 128 filters of 3x3 respectively, and 1 layer of convolutional layer using 4 filters of 1x1, all with Relu activation function, and then flatten the outputs to the 2 fully connected layers, and finally perform the SoftMax to output the recommended degree of each behavior according to the probability.

In the strategy neural network, the Relu function is chosen to be the activation function, and the specific definition of the Relu function is shown in Equation (5):

$$f(x) = \max(0, x) \tag{5}$$

II. C. 2) Value networks

The value network, is a deep learning neural network to score the input data. The main structure of the value network in this paper consists of 4 layers of convolutional neural network, the first 3 layers of convolutional neural network use 32, 64 and 128 3x3 filters, respectively, and in the 4th layer use 2 convolutional layers of 1x1 filters, and each convolutional layer uses the Relu activation function, and then flatten the output to the 3 fully-connected layers, and ultimately converge the output of the overall value of the rating.

II. C. 3) Risk networks

The principle of risk network implementation is to extract the features of the game situation through convolutional neural network, and then after multi-layer full connectivity, it is trained according to the data labels. In the data label contains all kinds of risk parameters, after a large amount of data training risk network model, it can be based on the input game situation to predict a variety of risk valuation, so that the gaming algorithm with the addition of risk network can predict the risk information. The specific number and type of risks can be set according to the actual application scenarios. The game system with added risk network is more realistic and instructive.

II. C. 4) Self-learning strategy value at risk network structure design

This paper proposes a self-learning strategy value risk network algorithm, which is mainly divided into two modules: self-game and strategy value risk network. The self-game module is responsible for generating training samples, and the strategy-value-risk network module is responsible for predicting strategies, risks and values.

Monte Carlo tree search algorithm is used for generating training samples, and the modified Monte Carlo tree search algorithm is used for cases where continuous behavior is required. Monte Carlo tree search algorithm in the selection of behaviors, the first multiple simulation and expansion operations will be calculated after the current state of the recommended value of each behavior, π represents the degree of recommendation of each behavior; in the end of the simulation phase of the Monte Carlo tree search algorithm, it can also be calculated in front of the risk exposure of each behavior, with D expressed in the current state of the risk faced; where S denotes the current in state; Z is an overall rating of the current state, which is only available at the end of a round of matchmaking, where victory is denoted by +1, defeat by -1, and a tie by 0; this constitutes the sample quaternion (S,π,D,Z).

The Strategy Value Risk Network records the behavioral recommendations and risks faced by each node of the Monte Carlo tree. With such a neural network, it is no longer necessary to construct a complete Monte Carlo tree to record the recommended values of all nodes and their behaviors; it is only necessary to construct a Monte Carlo subtree based on the current state. When constructing a Monte Carlo subtree node, the initial state of the node is set to the historical experience value, which can be obtained by querying the strategy value risk network so that the Monte Carlo subtree with historical experience can be reconstructed at any time.

III. Results and Discussion

III. A. Experimental design

III. A. 1) Simulation environment design

Since testing the model in a real business environment will reduce system usability and may bring unnecessary losses, this paper designs a simulation environment based on user logs. Firstly, a record is extracted from the user log in sequence, and then information such as user and target items are extracted from it, user history data is constructed as the input data for the enhanced recommendation model by combining the previously stored information, and finally an immediate return input model is computed based on the target items and the recommendation results to guide the further update, while the target items are added to the storage to update the user's history, until the user log is read.

III. A. 2) Experimental design

In order to verify the effectiveness of the enhanced recommendation model proposed in this chapter, four evaluation indexes, namely, accuracy, item diversity, MAP and NDCG, are selected, in which ILS mainly measures the diversity index.

Precision: indicates the ratio of the number of items that users really browse to the number of all recommended items in all the recommended results, the larger the value, the more accurate the recommendation results are, and the formula is as in equation (6):

$$precision@K = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{|R(u)|} \tag{6}$$

ILS: Indicates the degree of similarity of the recommended items in the recommendation list, the larger its value, the worse the diversity of the recommended results, the formula is as in equation (7):

$$ILS@K = \frac{2}{k(k-1)} \sum_{i \in L} \sum_{j \in L, j \neq i} S(i,j) \tag{7}$$

MAP: Consider the accuracy of sorting, user Top-N recommendation evaluation, the larger the value, the better the result. Defined as equation (8):

$$MAP@K = \frac{1}{m} \sum_{i=1}^{m} AP_u \tag{8}$$

NDCG: i.e. Normalized Discounted Cumulative Gain, is used to evaluate the users and their recommendation list results in the whole test set, this metric needs to calculate the DCG of each user first, and the DCG of a single user is defined in equation (9):

$$DCG_u = \sum_{i=1}^{k} \frac{2^{T(i)} - 1}{\log(i+1)} \tag{9}$$

The ideal DCG is then used to perform a normalization operation on the actual DCG as in Equation (10):

$$NDCG_u @ K = \frac{DCG_u}{IDCG_u} \tag{10}$$

III. B. Comparison of Depth MCTS and UCT Algorithms

III. B. 1) UCT algorithm

The UCT algorithm was used to guide the search, and after 30h of searching, the optimal solution of length 11 was obtained, and its decision sequence was [35, 95, 59, 31, 12, 28, 65, 45, 22], which indicated that the first 11 recommended targets were served by the recommended platforms in the decision sequence. Subsequent decision sequences can not be further obtained because the memory is full due to the need to store information on the nodes and other related information, and the search tree can not continue to search.

The node information of the decision result of length 11 is shown in Table 1, where W/N denotes the average Q value. In fact, there are nodes at each level with a smaller combined consumption value of time, but the search tree is simply oriented to node returns without empirical guidance, resulting in an inefficient algorithm when deeper nodes are not explored, and the search for the optimal solution is dependent on the arithmetic power, which needs to be relied on for continuous trial and error. So the efficiency of this UCT algorithm to find the solution is relatively low.

Table 1: Node information on UCT optimization decision

Serial number	Action	W/N
1	35	3.309
2	95	4.295
3	59	4.644
4	31	5.273
5	12	5.688
6	28	5.986
7	65	6.645
8	41	7.041
9	22	7.83
10	90	8.31
11	54	9.748

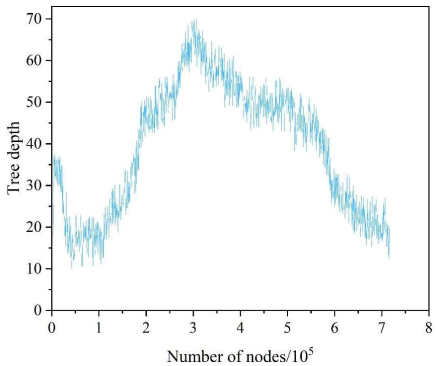


Figure 2: UCT search tree node depth

Figure 2 shows the structure of the search tree obtained using the UCT algorithm, where the burrs indicate that attempts were made to explore in all directions during the optimization process, but no better results were obtained. Despite the very large number of nodes, it still failed to find excellent results. From the figure, it can be seen that with the increase of depth, the nodes of the same layer are expanding, and UCT can only find the optimal solution by constant simulation, but this search method cannot effectively reduce the access of subtrees, and the

optimization search is inefficient. For the scaling problem, the UCT algorithm is less efficient and it is difficult to converge to a better solution.

III. B. 2) Deep MCTS

This algorithm sets the number of iterations to 30000 steps. It took 202.823s and obtained the sequence as [37, 53, 59, 15, 97, 33, 32, 12, 77, 83, 24, 43, 98, 72, 62, 89, 36, 30, 68, 39, 66, 17, 67, 58, 8, 2, 4, 43, 87, 56, 23, 79, 41, 84, 90, 28, 51, 10. 52, 71, 93, 69, 86, 6, 80, 48, 50, 5, 75, 82, 73, 47, 60, 42, 100, 25, 27, 64, 15, 76], which indicates that a total of 60 target recommendations are served by 60 digitally numbered recommendation platforms. The combined consumption value of the consumed time is obtained to be 5.974 in aggregate.

Table 2 shows the information of each node on the decision result of the improved MCTS algorithm. Where W/N denotes the average Q value. p is the selection probability of the corresponding action. The table also appears in the same layer of the search tree also exists in the competitive other nodes, the analysis is similar as in Experiment 1, and will not be repeated here. Because of space limitations, only the first 15 optimal solutions are shown here.

Table 2: Improved node information on the optimal decision of MCTS

Serial number	Action	W/N	p/%
1	37	6.021	81.26
2	53	6.242	72.32
3	59	6.943	77.43
4	15	7.131	83.36
5	97	7.604	88.31
6	33	8.709	95.4
7	32	9.193	94.63
8	12	9.717	94.95
9	77	10.843	96.06
10	83	11.221	94.34
11	24	11.955	96.62
12	43	12.323	93.08
13	98	12.716	94.83
14	72	13.447	93.44
15	62	14.196	96.27

The depth of the nodes of the improved MCTS search tree is shown in Fig. 3. The experimental results show the depth of all visited nodes in the search tree of the improved MCTS algorithm, and it can be seen that each search endeavors to expand the depth of the search tree, which is consistent with the task's goal of interest. There is very little burr formation in the search subtree, while the search is highly directional, indicating that the search can be used to efficiently find pathways to deeper subtrees. And the fact that each program loop can go deeper to a locally optimal predicted solution in a particular direction also suggests that subtrees at the same level, by selecting nodes with similar probabilities, have the same competitiveness.

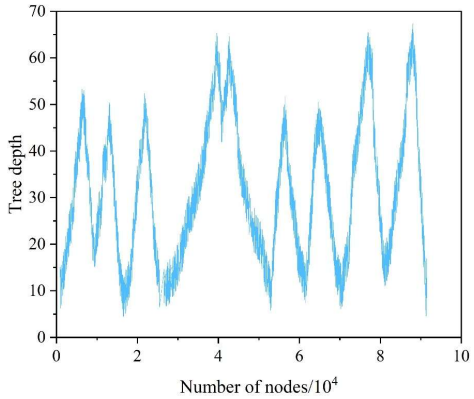


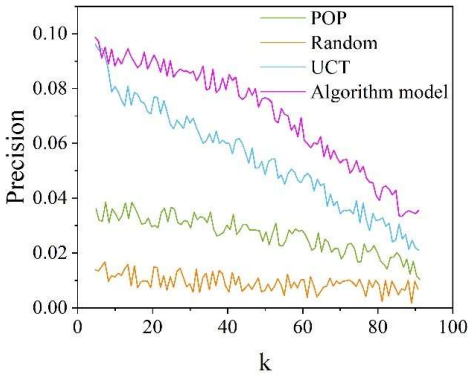
Figure 3: Improved MCTS search tree node depth

III. C. Experimental results and analysis

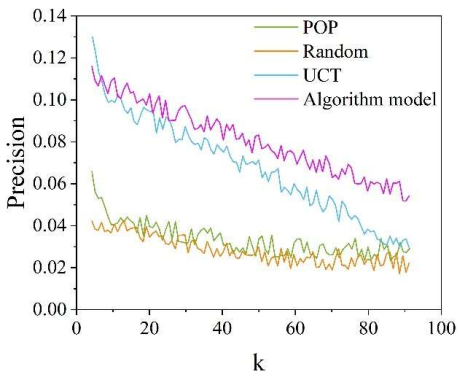
III. C. 1) Comparison of Top-N effects

For the UCT model, after the training is completed, the model calculates the sorting scores by calculating the user representations with all the item representations of the item library, and selects the first k items with the largest scores as the Top-N recommendation lists for each set of test data; in order to demonstrate the effect of the length of the recommendation lists on the performance of the model in a more accurate way, the recommendation performance of different k values is experimented, and the range of the value of k is set to = [5~90] in this experiment. Noting that the reinforced recommendation model is trained while obtaining the recommendation results, considering that the model training may be insufficient in the initial stage will affect the recommendation results, the dynamic Top-N results of the reinforced recommendation results are analyzed according to the different stages of the training data.

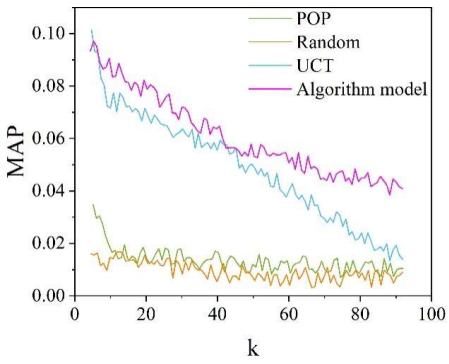
The experimental results are shown in Fig. 4, where the results in the left column indicate the test results on E-Commerce, and the right column indicates the test results on Retailrocker dataset, and the top, middle and bottom layers indicate the test results of the three index values of Precision, MAP, and NDCG, respectively, and the results show that, on the E-Commerce dataset, when the recommendation list length k=5, the reinforced recommendation model performs slightly worse than BPR in each evaluation index, and the rest of the performance is better than the other three comparative models; in addition to this, as the length of the recommendation list increases, all the models show a decreasing trend in the three indexes, and the self-learning strategic value risk network algorithm model outperforms the other comparative models in each index when the length of the recommendation list increases; in the Retailrocker dataset, the reinforced recommendation model is worse than the BPR model on the Precision metric when the recommendation list length k=5, and in the rest of the cases the reinforced recommendation model outperforms the other comparative models on the values of each metric; the recommendation metrics of all the models show a decrease when the recommendation list length is increased; among all the results, the random recommendation method is the worst, the UCT model outperforms the POP recommendation model and random recommendation model overall, and the self-learning strategy value risk network algorithm model recommendation model outperforms all comparison models overall.



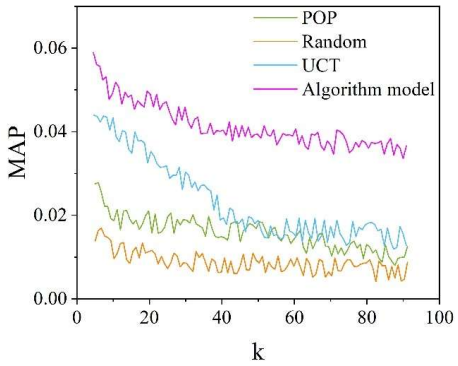
(a) Precision 1



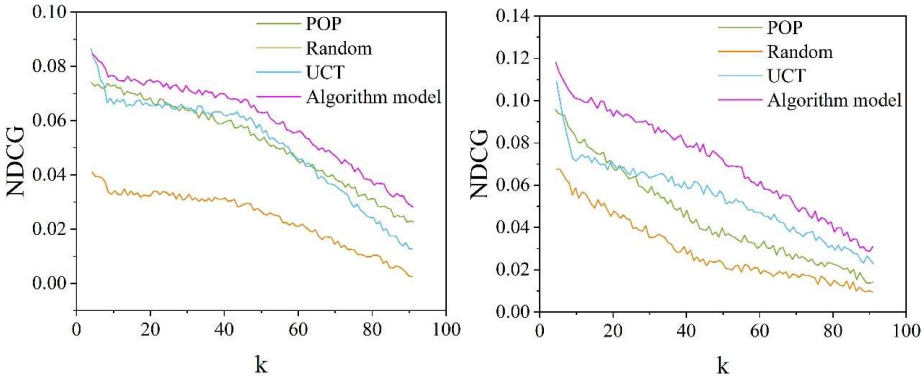
(b) Precision 2



(c) MAP 1



(d) MAP2



(e) NDCG 1 (f) NDCG 2

Figure 4: Top-N accuracy test results

In order to further illustrate the recommendation performance of the self-learning strategy value risk network algorithm model, Figure 5 shows the changes in the Precision indicator value of the reinforced model at different stages of the test set, as can be seen in the figure, as the training degree of the model continues to increase, the accuracy of the recommendation of the test data increases, in the initial training stage of the model, the model's recommendation accuracy is almost 0, and the final accuracy will reach 13%-22%.

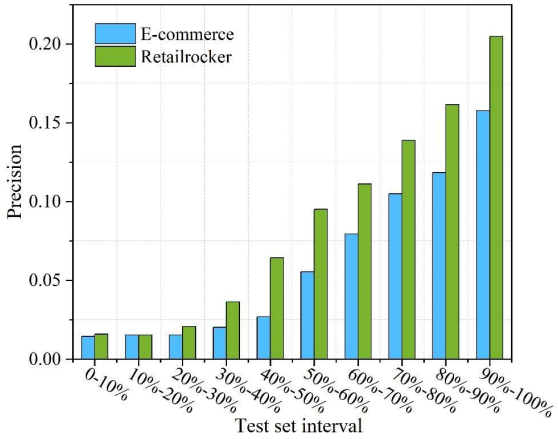


Figure 5: Precision (k=5) value dynamic test result

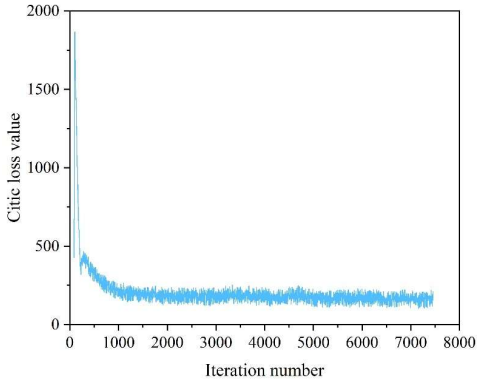


Figure 6: Convergence of loss value during critic training

As shown in Figure 6, after 1000 times of training, the loss function begins to converge, and Figure 7 shows the change of the immediate return value obtained by the Actor recommendation results of the model in the testing process, and it can be seen that with the increasing number of model training samples, the recommendation results

generated by the Actor will change in the direction of maximizing returns. This phenomenon shows that the self-learning strategy value risk network algorithm model needs enough training samples to achieve better recommendation effect.

The Top-N comparison experiments on each model demonstrate that the self-learning strategy value risk network algorithm model performs better in terms of recommendation accuracy, and that the recommendation performance improves further with the increase of model training data.

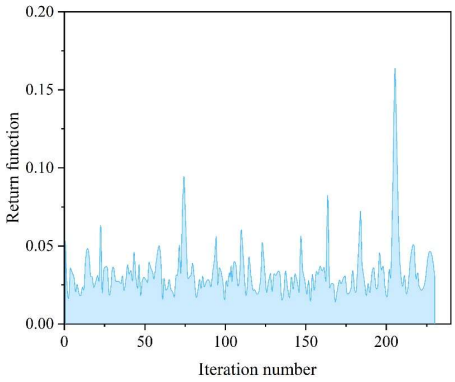
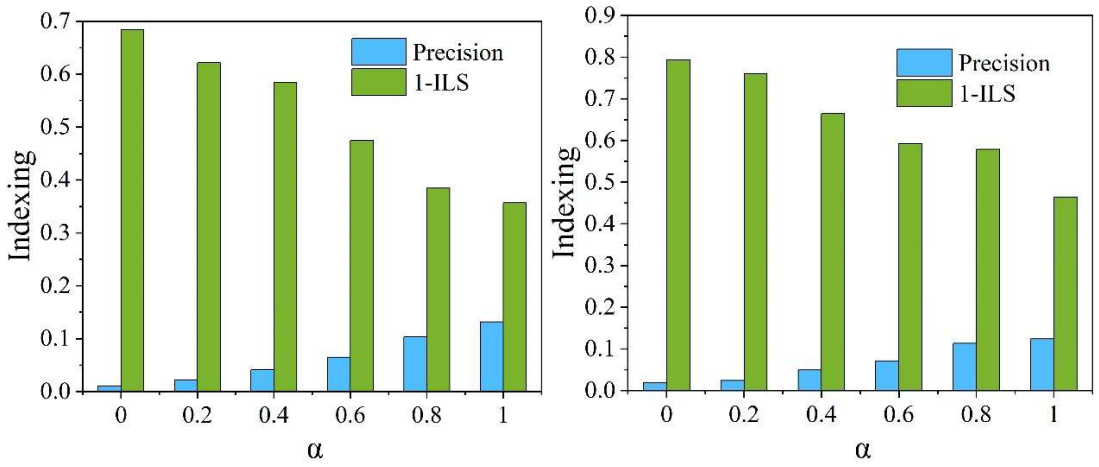


Figure 7: Changes in immediate reward value during actor testing

III. C. 2) Effect of reward function on recommendation results

In this paper, a special payoff function is designed to ensure that the goods within the recommended list are balanced between accuracy and diversity, this payoff function consists of a mixture of the sorting part and the diversity part, and the parameter α controls the weighting of the two parts, in order to validate the effect of different parameter settings on the results of the recommendation in this payoff function, the setting of α = [0.0, 0.2,0.4,0.6,0.8,1.0], with different values of α to train the self-learning strategy value risk network algorithm model recommendation model, the statistical accuracy index Precision and diversity index ILS, the experimental results are shown in Figure 8.

When the value of α increases, the accuracy indicator Precision of the self-learning strategy value risk network algorithmic model recommendation on both datasets increases, but the diversity of the recommendation list decreases, so it can be concluded that when α is small, the payoff function is biased towards accuracy, resulting in the self-learning strategy value risk network algorithmic model recommendation recommendation of the recommended The accuracy of the results is better and the categories of the recommendation results are more diverse, when α is larger, the reward function is biased towards diversity, resulting in better diversity of the recommendation results of the self-learning strategy value risk network algorithm model, but the accuracy will decrease.



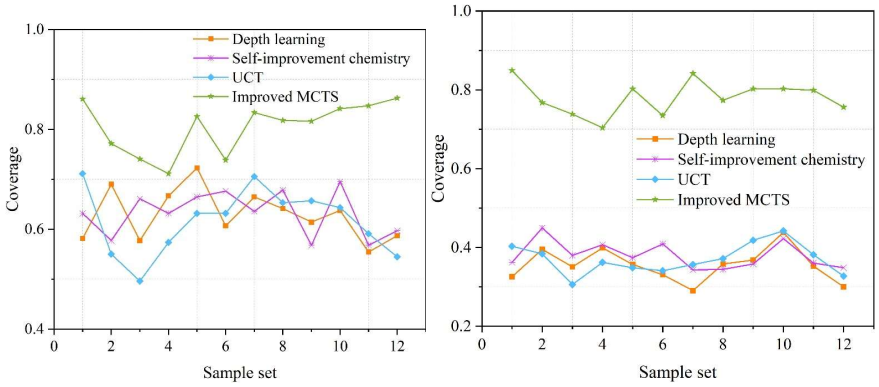
(a) E-Commerce data set (b) Retailrocker data set

Figure 8: The effect of the reward function on the recommendation

III. D. Recommendation Effect Analysis

III. D. 1) Coverage test results

The coverage test results of different methods under 100% test sample and 50% test sample are shown in Figure 9. It can be observed that there are differences in the performance of the 4 recommendation methods on different sample sets. No matter in which sample set, the coverage of the improved MCTS method is higher than that of the three methods, and the coverage of the deep learning method, the self-reinforcement learning method, and the UCT method under 50% test samples shows different degrees of decreases, while the coverage of the improved MCTS reinforcement recommendation method proposed in this paper shows a slight decrease, but the overall coverage is still high. Therefore, it shows that the method in this paper has obvious advantages in terms of coverage, and its performance is better than other comparative methods, which can cover the user group more comprehensively and provide a wider range of product recommendations.



(a) 100% test set (b) 50% test set

Figure 9: Coverage chart

III. D. 2) Diversity testing

The diversity test results of four different methods are shown in Table 3. Compared with deep learning methods, this paper's method performs better under different numbers of users, such as 0.9941 versus 0.8891 at user 10, and 0.9840 versus 0.9127 at user 20. Compared with the self-reinforcement learning recommendation method, the value of this paper's method under each number of users is much better than the other, 0.9705 vs. 0.4246 at user 30. Compared with the UCT recommendation method, the value is also higher under each number of users, which indicates that this paper's method has an outstanding performance in terms of diversity, and it can provide different users with better quality and more diversified e-commerce product recommendations.

Table 3: Diversity contrast

User	Depth learning	Self-improvement chemistry	UCT	This method
10	0.8891	0.6468	0.6784	0.9941
20	0.9127	0.5111	0.5673	0.9840
30	0.9115	0.4246	0.5117	0.9705
40	0.8295	0.3135	0.4006	0.9816
50	0.8846	0.5357	0.6228	0.9063

III. D. 3) Satisfaction test results

The results of the satisfaction tests of the four different methods are shown in Table 4. In each test, the satisfaction of this paper's method is higher than other recommendation methods. For example, in the 1st test, the satisfaction is 89.45%, while it is only 92.72% for the graph neural network and deep learning based method, 75.31% for the self-reinforcement learning recommendation method, and 79.44% for the UCT recommendation method. This advantage is even more obvious in the 8th test, which has a satisfaction rate of 98.94%, much higher than other methods. This indicates that the method of this paper can better meet the user needs in the product recommendation system of e-commerce platform, thus obtaining higher user satisfaction and more suitable for the recommendation of e-commerce products.

Table 4: Satisfaction test results(%)

Test frequency	Depth learning	Self-improvement chemistry	UCT	This method
1	71.47	75.31	79.44	92.72
2	73.80	77.53	81.78	95.61
3	72.44	76.18	78.55	94.36
4	74.91	79.84	82.33	96.83
5	70.46	74.20	77.44	92.03
6	72.69	78.64	80.77	97.59
7	72.02	75.87	79.22	93.53
8	76.02	80.86	83.03	98.94
9	71.33	74.87	78.11	93.14
10	71.47	75.31	79.44	92.72

These data results not only show that this paper's method is able to provide richer and more diverse product recommendations, but also reflect its ability to satisfy users' diverse needs. This paper's method is able to capture users' personalized preferences more effectively, avoiding the singularity and repetitiveness of the recommendation results, and thus enhancing users' satisfaction and loyalty. The method in this paper not only excels in diversity, but is also more superior in overall performance.

IV. Conclusion

In this paper, deep learning is combined with Monte Carlo tree search algorithm to propose the improved MCTS algorithm, on the basis of which the improved MCTS algorithm is fused to design the algorithm recommendation model for self-learning strategy value risk network. The experimental results prove that the improved MCTS algorithm has high comprehensive performance in both solution speed as well as solution quality, and the algorithmic recommendation of the self-learning strategy value risk network outperforms other models in Top-N effect comparison, and has better recommendation effect. Finally, the improved MCTS algorithm is analyzed for its recommendation effect, and the algorithm has achieved remarkable results in terms of diversity of recommendation results and user satisfaction, with the diversity peak as high as 0.9941 and the user satisfaction reaching 97.59%.

References

[1] Huseynov, F., & Özkan Yıldırım, S. (2019). Online consumer typologies and their shopping behaviors in B2C e-commerce platforms. Sage Open, 9(2), 2158244019854639.

[2] Jeong, H., Yi, Y., & Kim, D. (2022). An innovative e-commerce platform incorporating metaverse to live commerce. International Journal of Innovative Computing, Information and Control, 18(1), 221-229.

[3] Alamdari, P. M., Navimipour, N. J., Hosseinzadeh, M., Safaei, A. A., & Darwesh, A. (2020). A systematic study on the recommender systems in the E-commerce. Ieee Access, 8, 115694-115716.

[4] Shrivastava, R., Sisodia, D. S., & Nagwani, N. K. (2024). Deep ensembled multi-criteria recommendation system for enhancing and personalizing the user experience on e-commerce platforms. Knowledge and Information Systems, 66(12), 7799-7836.

[5] Bag, S., Ghadge, A., & Tiwari, M. K. (2019). An integrated recommender system for improved accuracy and aggregate diversity. Computers & Industrial Engineering, 130, 187-197.

[6] Abdul Hussien, F. T., Rahma, A. M. S., & Abdulwahab, H. B. (2021). An e-commerce recommendation system based on dynamic analysis of customer behavior. Sustainability, 13(19), 10786.

[7] Zhu, Z., Wang, S., Wang, F., & Tu, Z. (2022). Recommendation networks of homogeneous products on an E-commerce platform: Measurement and competition effects. Expert Systems with Applications, 201, 117128.

[8] Zhang, T. (2024, August). A Research Proposal for Promoting Fairness in E-Commerce Recommendation Systems. In 2024 11th International Conference on Behavioural and Social Computing (BESC) (pp. 1-5). IEEE.

[9] Yang, F. (2023). Optimization of personalized recommendation strategy for e-commerce platform based on artificial intelligence. Informatica, 47(2).

[10] Li, S. S., & Karahanna, E. (2015). Online recommendation systems in a B2C E-commerce context: a review and future directions. Journal of the association for information systems, 16(2), 2.

[11] Shang, F., Shi, J., Shi, Y., & Zhou, S. (2024). Enhancing e-commerce recommendation systems with deep learning-based sentiment analysis of user reviews. International Journal of Engineering and Management Research, 14(4), 19-34.

[12] Gogna, A., & Majumdar, A. (2017). DiABIO: Optimization based design for improving diversity in recommender system. Information Sciences, 378, 59-74.

[13] Salampasis, M., Katsalis, A., Siomos, T., Delianidi, M., Tektonidis, D., Christantonis, K., ... & Diamantaras, K. (2023). A Flexible Session-Based Recommender System for e-Commerce. Applied Sciences, 13(5), 3347.

[14] Wu, Y., & Yusof, Y. (2024). Emerging Trends in Real-time Recommendation Systems: A Deep Dive into Multi-behavior Streaming Processing and Recommendation for E-commerce Platforms. J. Internet Serv. Inf. Secur, 14(4), 45-66.

[15] Gharahighehi, A., & Vens, C. (2021). Personalizing diversity versus accuracy in session-based recommender systems. SN Computer Science, 2(1), 39.

[16] Sainiº, K., & Singh, A. (2023). Hybrid recommender system for e-commerce: A comprehensive. Journal of Harbin Engineering University, 44(8).
[17] Khodabandehlou, S. (2019). Designing an e-commerce recommender system based on collaborative filtering using a data mining approach. International Journal of Business Information Systems, 31(4), 455-478.
[18] Li, C. (2024). A Personalized Product Recommendation System for E-Commerce Platforms Based on Artificial Intelligence and Image Processing Technologies. Traitement du Signal, 41(6).
[19] Piao, J., Zhang, G., Xu, F., Chen, Z., Zheng, Y., Gao, C., & Li, Y. (2021). Bringing friends into the loop of recommender systems: an exploratory study. Proceedings of the ACM on Human-Computer Interaction, 5(CSCW2), 1-26.
[20] Świechowski, M., Godlewski, K., Sawicki, B., & Mańdziuk, J. (2023). Monte Carlo tree search: A review of recent modifications and applications. Artificial Intelligence Review, 56(3), 2497-2562.
[21] Vodopivec, T., Samothrakis, S., & Ster, B. (2017). On monte carlo tree search and reinforcement learning. Journal of Artificial Intelligence Research, 60, 881-936.
[22] Rossi, L., Winands, M. H., & Butenweg, C. (2022). Monte Carlo Tree Search as an intelligent search tool in structural design problems. Engineering with Computers, 38(4), 3219-3236.
[23] Haeri, S., & Trajković, L. (2017). Virtual network embedding via Monte Carlo tree search. IEEE transactions on cybernetics, 48(2), 510-521.
[24] Kaufmann, E., & Koolen, W. M. (2017). Monte-Carlo tree search by best arm identification. Advances in Neural Information Processing Systems, 30.
[25] Cheng, S., Kandemir, M. T., & Hong, D. Y. (2024). Speculative monte-carlo tree search. Advances in Neural Information Processing Systems, 37, 88664-88683.
[26] Nazareth, D. L., Choi, J., & Ngo-Ye, T. (2024). Investing in security-as-a-service for e-commerce infrastructure by small and medium enterprises: a Monte Carlo approach. Journal of Systems and Information Technology, 26(2), 257-275.
[27] Ma, T. M., Wang, X., Zhou, F. C., & Wang, S. (2023). Research on diversity and accuracy of the recommendation system based on multi-objective optimization. Neural Computing and Applications, 35(7), 5155-5163.
[28] Qi, L., Xu, X., Zhang, X., Dou, W., Hu, C., Zhou, Y., & Yu, J. (2016). Structural balance theory-based e-commerce recommendation over big rating data. IEEE Transactions on Big Data, 4(3), 301-312.
[29] Kim, J., Choi, I., & Li, Q. (2021). Customer satisfaction of recommender system: Examining accuracy and diversity in several types of recommendation approaches. Sustainability, 13(11), 6165.
[30] Hussien, F. T. A., Rahma, A. M. S., & Wahab, H. B. A. (2021, May). Recommendation systems for e-commerce systems an overview. In Journal of Physics: Conference Series (Vol. 1897, No. 1, p. 012024). IOP Publishing.
[31] Zanon, A. L., da Rocha, L. C. D., & Manzato, M. G. (2022). Balancing the trade-off between accuracy and diversity in recommender systems with personalized explanations based on linked open data. Knowledge-Based Systems, 252, 109333.
[32] Yu, T., Guo, J., Li, W., Wang, H. J., & Fan, L. (2019). Recommendation with diversity: An adaptive trust-aware model. Decision Support Systems, 123, 113073.
[33] Gogna, A., & Majumdar, A. (2017). Balancing accuracy and diversity in recommendations using matrix completion framework. Knowledge-Based Systems, 125, 83-95.
[34] Zhang, L., Wei, Q., Zhang, L., Wang, B., & Ho, W. H. (2020). Diversity balancing for two-stage collaborative filtering in recommender systems. Applied Sciences, 10(4), 1257.
[35] Xie, R., Liu, Q., Liu, S., Zhang, Z., Cui, P., Zhang, B., & Lin, L. (2021). Improving accuracy and diversity in matching of recommendation with diversified preference network. IEEE Transactions on Big Data, 8(4), 955-967.
[36] Ping, Y., Li, Y., & Zhu, J. (2024). Beyond accuracy measures: the effect of diversity, novelty and serendipity in recommender systems on user engagement. Electronic Commerce Research, 1-28.
[37] Wu, S., Kou, H., Lv, C., Huang, W., Qi, L., & Wang, H. (2020). Service recommendation with high accuracy and diversity. Wireless Communications and Mobile Computing, 2020(1), 8822992.
[38] Zaizi, F. E., Qassimi, S., & Rakrak, S. (2024, April). Enhancing E-commerce Recommender Systems through Multi-Objective Immune Algorithm. In Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security (pp. 1-1).
[39] Milán Hugyi. (2025). Practical Usefulness of the Monte-Carlo Method in Processes of the Automotive Industry. Land Forces Academy Review,30(1),141-147.