

Machine learning algorithms in the development of recommender systems

Naihao Guan

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu, 221116, China

12201310@cumt.edu.cn

Abstract. With the continuous development and widespread use of the Internet, individuals often find themselves overwhelmed by the vast amount of information available. To address this issue and better cater to users' personalized needs, recommender systems have emerged. A recommender system is a technology that provides users with customized content or services based on their preferences and interests. Its primary goal is to predict user behavior as accurately as possible in order to recommend relevant items. Machine learning algorithms play a crucial role in the functioning of recommender systems. These algorithms not only automatically analyze and process large volumes of data but also extract valuable features from historical data, enabling accurate predictions for unknown data. The rapid progress of deep learning technology has significantly enhanced the accuracy and efficiency of recommender systems. Deep learning enables recommender systems to better grasp user behavior, interests, and preferences, leading to more precise predictions of user needs and behaviors. Furthermore, deep learning can handle unstructured data such as images, audio, and text, extracting valuable features to improve recommendations. In the future, as machine learning and artificial intelligence technologies continue to advance and gain popularity, recommender systems will find broader applications. Apart from the traditional e-commerce sector, recommender systems can also be applied to social networks, news media, online education, and other fields. Simultaneously, in the research and implementation of recommender systems, due attention must be given to user privacy protection and data security. Only by ensuring user privacy and data security can recommender systems effectively meet user needs and gain wider acceptance.

Keywords: Machine Learning, Recommender System, Artificial Intelligence.

1. Introduction

Recommender systems, as an important technology, play a crucial role in the Internet era. It provides personalised recommendation information by analysing users' historical behaviours and personal characteristics, in order to solve problems such as information overload and selection difficulties. However, in the face of ever-growing massive information resources [1], traditional recommendation algorithms face challenges such as data sparsity, cold start, and recommendation accuracy.

In recent years, the rapid development of machine learning algorithms and the wide application of big data have brought new opportunities to the field of recommender systems. Machine learning algorithms can achieve more accurate personalised recommendations by learning users' historical behaviour and interest models. It improves the degree of automation and efficiency of recommendation

systems with the help of large-scale data learning and training. In addition, combined with technologies such as deep learning, machine learning algorithms can mine users' potential interests and hidden association rules to further optimise recommendation results.

However, the application of machine learning algorithms in recommender systems also faces a series of challenges. First, data sparsity and cold-start problems limit the accuracy and degree of personalisation of traditional algorithms. Second, the explanatory and interpretable nature of the recommendation results is under scrutiny, and the problem of how to explain the reasons and rationale for recommendations to users' needs to be addressed. In addition, with the increasing awareness of personal privacy protection, how to strike a balance between personalised recommendation and privacy protection is also an important topic.

Therefore, it is of great significance to conduct in-depth research on the application of machine learning algorithms in recommender systems. This can not only promote the development and innovation in the field of recommender systems, but also provide guidance for the design and optimisation of recommender systems in practical applications. By exploring ways to solve the above challenges, the accuracy, degree of personalisation and user satisfaction of recommender systems could be further improved, and provide users with more intelligent and customised recommendation services.

The application of machine learning algorithms in the development of recommender systems has important academic background and practical significance. Through in-depth research, this work can overcome the challenges faced by recommender systems, improve their performance and effectiveness, bring feasible solutions to the problem of information overload, and promote academic and technological progress in related fields.

This paper will present the evolution of recommender systems and the algorithms and techniques used at different stages. It will first introduce the early traditional recommender systems, including collaborative filtering-based and content filtering-based approaches. Then it will introduce the application of machine learning in recommender systems, and introduce the improvement of model-based collaborative filtering algorithms and content filtering algorithms. Then the application of deep learning in recommender systems will be introduced in detail, including the recommendation models based on neural networks and attention mechanisms. Finally, the application of reinforcement learning in recommender systems and the development trend of hybrid recommender systems will be discussed, and the future development direction of recommender systems will be looked forward to.

2. Conventional Recommendation Algorithms

2.1. Content-based Recommendation Algorithms

Early traditional recommendation algorithms mainly include content-based recommendation algorithms and collaborative filtering algorithms. These algorithms implement personalised recommendations in different ways to provide users with items that match their interests and preferences.

Content-based recommendation algorithms are a method of making recommendations based on the attribute characteristics of items and the user's preferences. It builds feature vectors of items by analysing their attributes, such as titles, descriptions, labels, etc., and calculates the similarity between different items. When a user expresses his/her preference or is interested in an item, the system can recommend other items with high similarity to the user's existing interest to the user based on the user's preference and historical behaviour. Content-based recommendation algorithms can accurately match the user's interests, and are especially suitable for small-scale datasets and situations where the attributes of items are clear.

Collaborative filtering algorithms, on the other hand, are methods for recommendation based on user behavioural data or item relevance [2]. It discovers the similarity of users' interests by analysing the behavioural similarity between users or the correlation between items. The user-based collaborative filtering algorithm will find other users with the most similar interests to the target user based on the user's behavioural records, and then based on the preferences of these similar users, the items they like will be recommended to the target user. The item-based collaborative filtering algorithm, on the other

hand, recommends similar items to the user's favourite items by calculating the similarity between the items. Collaborative filtering algorithms can solve the problems of cold-start and data sparsity, and are suitable for large-scale datasets and situations where it is difficult to extract item attributes directly.

These early traditional recommendation algorithms achieved personalised recommendation based on information such as content and user behaviour, using similarity calculation and correlation analysis [3]. They provide important ideas and foundations for the development of subsequent recommendation algorithms and have a wide range of applications in the field of recommender systems.

2.2. Analysis of Conventional Recommendation Algorithms

Traditional recommendation algorithms have some inherent shortcomings in their early stages of development that limit the accuracy and user experience of recommendation systems. First, the cold-start problem is a common challenge. Traditional algorithms often rely on users' historical data to make recommendations, but the lack of sufficient personal information and behavioural data is an obstacle for new users or just-launched recommender systems. As a result, new users are often unable to enjoy personalised recommendation services, which can affect user satisfaction.

Second, data sparsity is also a drawback of traditional recommendation algorithms. User behavioural data tends to be sparse, i.e., the user only evaluates or interacts with a small number of items without explicit feedback on other items. This sparseness of data makes it difficult for traditional algorithms to accurately capture users' interests and preferences, which leads to inaccurate recommendation results.

In addition, the limited feature representation capability of traditional recommendation algorithms is also one of the problems. Traditional algorithms usually use simplified feature representations to describe users and items, based on tags or keywords. However, such simplified feature representations may not adequately capture the complex features and relationships of users and items. As a result, when dealing with complex recommendation scenarios, traditional algorithms are limited in their feature representations and thus cannot provide accurate personalised recommendations.

Finally, early traditional recommendation algorithms lack real-time performance. Due to the use of offline batch processing for computation and recommendation, the real-time nature of recommendation results is poor. However, with the rapid development of the Internet and mobile applications, the increase in users' demand for real-time makes the traditional algorithms lagging and unable to meet users' immediate recommendation needs.

To summarise, early traditional recommendation algorithms have shortcomings such as cold-start problem, data sparsity, limited feature expression capability and lack of real-time performance [4]. These problems limit the performance of traditional recommendation systems in terms of accuracy and user experience. With the development and application of machine learning algorithms, recommender systems have made significant progress, and the next section will focus on the application of machine learning algorithms in recommender systems.

3. Machine Learning-based Recommendation Algorithms

3.1. User-Item Matching

Collaborative filtering algorithms with the introduction of machine learning improve the accuracy and effectiveness of traditional recommender systems. Machine learning algorithms are better able to capture user preferences and provide personalised recommendations by learning user interests and item characteristics from large-scale data.

User-item matching algorithm is an algorithm that matches users with items that fit their needs by analysing their needs and item characteristics. Its implementation usually includes the following steps: first, collect user demand information, which can be obtained through user input, questionnaires, etc., to obtain the specific needs and preferences of users; then, collect information about the characteristics of the project, including the type of the project, the size, and the skills required; next, calculate the matching score by evaluating the degree of match between user demand and project characteristics [5]. According to the matching score, the projects are sorted according to the match score and filtered according to the

user's filtering conditions in order to present the most suitable projects to the user; finally, the matching algorithm is continuously optimised according to the user's feedback and behaviour to improve the accuracy and personalisation. User-project matching algorithms can be applied in various fields to help users quickly find projects that meet their needs, improve user satisfaction and efficiency, and also help project owners more accurately find their target audience.

After the introduction of machine learning algorithms, the collaborative filtering algorithm can be roughly divided into 4 steps. First, for each item to be analysed, some key features need to be extracted from it, which can be text descriptions, keyword tags, attribute values, etc. of the content. For example, for a film recommendation system, features such as the director, actors, genre, and plot of the film can be extracted. Secondly, the user's interest is modelled, which is usually represented by using the user's historical behavioural data. These behavioural data will be transformed into a vector of user features, where each feature indicates the degree of user preference for a particular content. Then, the similarity between items and user features is calculated. Some similarity measures such as cosine similarity, Euclidean distance or Pearson's correlation coefficient are usually used to measure the degree of similarity between user feature vectors and item feature vectors. Finally, based on the computed similarity, the item most relevant to the user's interest is selected as the recommendation result. It is possible to set a threshold to filter out items with similarity higher than the threshold, or to select a certain number of items with top similarity ranking.

3.2. Clustering-based Solution

Clustering algorithm is an unsupervised learning method commonly used in data mining and machine learning. Its goal is to group data points with similar characteristics into clusters to reveal patterns and structure in the data. The basic idea of clustering algorithms is to form clusters based on a similarity measure between data points, where data points within clusters should be similar and data points between different clusters should be significantly different.

There are several common approaches to clustering algorithms. Among them, the K-means algorithm is a classical clustering algorithm that iteratively calculates the distance between data points and the centre of mass and assigns the data points to the cluster represented by the closest centre of mass. The centre of mass is then recalculated for each cluster until the position of the centre of mass no longer changes or a predetermined number of iterations is reached [6].

Another common clustering algorithm is the hierarchical clustering algorithm, which progressively merges or splits data points to construct a hierarchical clustering result. This algorithm can construct a clustering tree or a clustering graph based on similarities or distances between data points [7].

There is another class of clustering algorithms known as density clustering algorithms, which define clusters as regions of high density and discover the cluster structure based on the density of the data points. DBSCAN and OPTICS are typical examples of density clustering algorithms. There are also clustering algorithms based on probabilistic models, such as Gaussian mixture models, which assume that the data conforms to a certain probability distribution and assign data points to different clusters by adjusting the model parameters [7].

Clustering algorithms have a wide range of applications in many fields, such as market segmentation, social network analysis, image segmentation and anomaly detection. Through clustering algorithms, hidden patterns and structures can be discovered from complex data to provide support and insight for decision making.

3.3. Association Rule Algorithm

Association rule algorithm is a data mining method for finding relationships between itemsets from large-scale datasets. Usually, these itemsets include multiple attributes or features, and there is some kind of association between different itemsets. By extracting these association rules, important patterns and trends could be identified in the data, discover hidden knowledge and perform predictive analyses and so on.

The core idea of association rule algorithm is to find association rules based on frequent itemsets. The frequent itemsets are filtered by setting a minimum support threshold. Then, the strength of the association rules is determined based on the confidence level, which indicates the probability of the simultaneous occurrence of itemset A and itemset B divided by the probability of the occurrence of itemset A. If the confidence level is higher than a predefined threshold, the two itemsets are associated and can be defined as an association rule.

Common association rule algorithms include Apriori algorithm, FP-Growth algorithm and so on. Among them, Apriori algorithm is an algorithm based on generating candidate itemsets and pruning. It generates new candidate sets by constantly merging frequent item sets, and then uses support and pruning techniques to filter out frequent item sets [8]. FP-Growth algorithm, on the other hand, is an efficient algorithm based on the data structure FP-Tree, which can discover frequent item sets faster [9].

Association rule algorithms have many applications such as shopping basket analysis, recommender systems, marketing, etc. For example, in shopping basket analysis, association rule algorithms could be leveraged to explore the relationship between consumers' purchases in order to develop promotional strategies for specific products. In recommender systems, association rule algorithms could be used to recommend similar products or services to improve user satisfaction.

It should be noted that association rule algorithms also have some problems and difficulties. Firstly, if the dataset to be mined is too large, computing frequent itemsets and association rules will become very time-consuming, which requires the use of efficient algorithms and techniques. Secondly, due to data complexity and noise interference, some association rules may be inaccurate or missing, and thus need to be evaluated for accuracy and reliability.

4. Deep Learning-based Recommendation Algorithms

4.1. Preliminary Knowledge

A neural network is a computational model inspired by the human nervous system for modelling and processing complex non-linear relationships. It consists of many simple processing units (neurons) with weighted connections that process and learn information by passing signals.

Neural networks were first introduced to solve problems that traditional computers could not effectively handle. Unlike traditional rule-based programming methods, neural networks learn and determine the relationship between inputs and outputs by training data sets to automatically discover patterns and regularities in the data [10].

When sufficiently trained, neural networks are able to accurately predict and classify new unseen inputs. This is because during training, the neural network automatically learns the mapping relationship between inputs and outputs by adjusting the connection weights. By increasing the number of layers and number of neurons, neural networks can handle more complex problems and large-scale data sets.

4.2. Convolutional Neural Network (CNN)

Convolutional neural networks have a wide range of applications in recommender systems. Traditional recommender systems mainly rely on methods such as collaborative filtering to make recommendations, but these methods ignore the content information of items. CNN, on the other hand, as a powerful image and sequence data processing model, can effectively capture the features and similarities of the items to improve the accuracy and personalisation of the recommendations.

CNN adds a convolutional layer, pooling layer, and fully connected layer to the lifted neural network to ensure accuracy [11]. The convolutional layer is used to extract local features from the input data. Convolutional operations are performed on the input data by defining a series of convolutional kernels. Each convolution kernel captures different features in the input data such as edges, texture, etc. The convolution operation slides over the input data and performs elementwise product and summation operations to generate a feature mapping. With multiple different convolution kernels, multiple different feature mappings can be obtained to extract richer feature representations.

The pooling layer is mainly used to reduce the size of the feature mapping and retain important feature information. Common pooling operations include maximum pooling and average pooling. Through pooling operations, the number of network parameters and computation can be reduced, while maintaining the positional invariance of features and improving the generalisation ability of the model.

The fully connected layer flattens the high-level features after convolution and pooling, and performs matrix multiplication operations with the weights, plus bias terms, and performs nonlinear transformations through the activation function. The fully connected layer captures the global information in the input data and generates the final output. It is commonly used for learning and predicting interaction features between users and items, such as rating prediction, sorting recommendation and other tasks.

4.3. Recurrent Neural Network

Recurrent Neural Networks have a wide range of applications in recommender systems. RNN is a neural network model suitable for processing sequential data, which is able to process the current input by remembering previous information. In recommender systems, RNNs are able to capture sequences of user behaviours and temporal features of items to improve the accuracy and personalisation of recommendations.

RNNs are particularly suitable for processing sequential data, such as historical sequences of user behaviour [12]. By taking the user's historical behavioural sequences as input, RNNs can learn the temporal dependencies between the sequences and predict items or behaviours that may be of interest to the user in the future. This enables personalised sequence prediction recommendations. RNN can be used in personalised sequencing models. By feeding the user's historical behavioural sequences into the RNN, the model can learn the evolution of the user's interests and use this information to score or rank candidate items. This can provide users with more personalised and accurate recommendation results.

For conversational recommendation tasks, RNNs can be used to capture the user's interest shifts and contextual changes during the session [13]. By taking the user's session history as input, the RNN can make predictions about the next possible behaviour or session. This allows for real-time recommendations of items that match the user's intent during the session.

RNNs can also be used to generate personalised recommendation content. By taking a sequence of a user's historical behaviours as input and using an RNN model to generate a new sequence of items or text recommendations. This can provide the user with a personalised recommendation experience for songs, movies, articles, etc.

4.4. Attention Mechanism

By introducing the attention mechanism, the model can weight different parts of the input sequence based on the current contextual information. Specifically, the attention mechanism computes an attention weight vector to represent how much attention the model pays to different inputs. This weight vector can vary depending on different tasks and model architectures, but is usually obtained by calculating the similarity between the inputs and the current model state. The model then weights and sums the different parts of the input sequence according to the attention weight vector to generate the final output or further processing.

Using the attention mechanism, the model can process the sequence data more flexibly by focusing more attention on the parts that are relevant to the task at hand. This mechanism allows the model to understand the input data more accurately and better predict, classify, or generate.

The model usually consists of a multi-layer neural network with strong expressive power [14]. It can extract higher level feature representations through multiple layers of hidden units to fully capture the associative relationships between users and items. This enables the model to better understand and mine the underlying patterns and user behavioural patterns in the data. The adaptive attention model can dynamically adjust the level of attention to different items based on the user's interest and importance. This enables the model to better capture users' personalised preferences and generate more accurate recommendation results. In addition, with dynamic learning capabilities, it can flexibly adjust to changes

in different users and different scenarios. This means that the model can adaptively update the user's interest representation and attention weight with changes in user behaviour, solving the long-tail problem thus continuously providing personalised recommendation services. Through the adjustment of attention, it helps users filter and recommend personalised information, alleviating the problem of information overload. It improves the user experience by placing the most relevant and valuable items in front of the user's eyes based on their preferences.

5. Reinforcement Learning-based Recommendation Algorithms

5.1. Deep Q-Learning (DQN)

DQN in recommender systems is a technique that uses deep reinforcement learning methods to solve recommendation problems. It combines deep neural networks and Q-learning algorithms to guide the decision-making process of a recommender system by learning a Q-value function.

In recommendation systems, DQN can be applied in several ways. Firstly, DQN uses deep neural networks to represent the state of users and items, using feature vectors as inputs and transforming them through a multilayer network to obtain a representation of the state. Secondly, the goal of DQN is to learn a Q-value function that is used to estimate the value of taking different actions in a given state. The recommender system can select behaviours with the highest Q-value based on the current state, such as the items recommended to the user or the selection of recommendation strategies.

In order to solve the problems of sample correlation and unstable data distribution, DQN introduces an empirical playback mechanism. This means that the user's observation sequences, behaviours taken, rewards obtained, and next states are saved into the experience pool when interacting with the user [15]. Then, batch samples are randomly selected from the experience pool for training to reduce the correlation between samples.

To enhance the stability of the algorithm, DQN also introduces a goal network. The target network is a copy of the main network structure whose parameters remain constant over time. By using the Q-value estimation of the target network to update the parameters of the main network, DQN reduces the volatility of the target value and improves the stability of training.

5.2. Actor-Critic

The Actor-Critic approach in recommender systems is a technique widely used in reinforcement learning frameworks to optimise recommendation policies by using both Actor and Critic components. This approach combines the advantages of policy gradient and value function estimation to achieve better performance in recommender systems [16].

In the Actor-Critic approach, the Actor is responsible for decision making, i.e., selecting a behaviour as a recommendation outcome based on the current state. The Actor guides the recommendation decision by learning a policy function, which can be represented as a neural network in the recommender system with user features and item features as inputs. Based on this policy function, the Actor can output the probability distribution of a recommended behaviour or a set of recommended behaviours.

Critic, on the other hand, is responsible for evaluating the Actor's decisions, and it estimates the value of a recommended behaviour by learning a value function. The value function measures how good or bad the recommended behaviour is and provides immediate feedback signals. In a recommender system, Critic's value function can be used to assess user satisfaction with a recommendation or to predict the quality of a recommendation.

5.3. Proximal Policy Optimization (PPO)

PPO is an optimisation algorithm widely used in reinforcement learning frameworks for training and optimising policy functions in recommender systems. PPO uses a proximal policy optimisation approach with high stability and convergence performance.

The core idea of PPO is to ensure that the learning process is stable and converges to a better policy by limiting the magnitude of change in the policy function at each update. This limitation is achieved

by using a mechanism of proportional pruning of importance, such as using the Cramer-Landau distance or relative entropy to measure the difference between the old and the new strategies. The training process of a PPO algorithm usually consists of the following steps [17]: collecting training data, interacting with the environment using the current policy function, collecting user feedback, reward signals, and state transfer information; calculating the dominance estimation, and estimating each state action using a Critic network to estimate the dominance of each state-action pair, the dominance of the current strategy over the expected payoff of the reference strategy; updating the strategy function by maximising the objective function, which usually consists of two parts: a strategy gradient term and a regularisation term. The strategy gradient term is computed based on the dominance estimate and the probability distribution of the current strategy, and the regularisation term is used to limit the magnitude of the strategy change, thus controlling the stability of the training process; the update is performed again using the new data collected until a predefined number of training rounds or convergence conditions are reached.

5.4. Thompson Sampling

Thompson sampling in recommender systems is a probabilistic algorithm used to solve the multi-armed gambling machine problem and is also widely used in recommender systems. It is based on the idea of Bayesian inference, which optimises the recommendation strategy by balancing the trade-off between exploration and exploitation through continuous exploration and exploitation.

In recommender systems, the multi-armed gambling machine problem can be viewed as the problem of selecting and allocating resources to maximise the cumulative returns when given a set of recommendation options. Each recommendation option can be viewed as a gambling machine, and it is required to select the best gambling machine and make recommendations under uncertainty.

The core idea of Thompson's sampling algorithm is to model uncertainty by establishing a probability distribution for each recommendation option and using Bayesian inference to update the probability distribution [18]. By setting up an initial probability distribution for each recommendation option. Then a sample is sampled from the probability distribution of each recommendation option to obtain a set of possible recommendation outcomes. For each sample, its corresponding cumulative return is calculated based on known observations. The probability distribution of each recommendation option is then updated based on the observations, using a Bayesian inference approach to update the parameters. Sampling, evaluation and updating are repeated to continuously adjust the probability distribution. Finally, the best recommendation option is selected for recommendation based on the updated probability distribution.

6. Conclusion

The aim of this paper is to explore the application of machine learning algorithms in recommender systems and to develop the discussion by introducing early traditional recommendation algorithms, the introduction of machine learning, deep learning models and reinforcement learning models. First, traditional recommendation algorithms are reviewed, including content filtering and collaborative filtering-based approaches. With the rise of big data and the development of machine learning techniques, researchers began to introduce machine learning algorithms into recommender systems to improve their accuracy and personalisation. Then, different applications of machine learning algorithms in recommender systems are introduced. Deep learning models have become a hot research topic to mine more complex relationships between users and items by building multi-layer neural networks. These models can learn more abstract feature representations from massive data to provide more accurate recommendation results. In addition, the application of reinforcement learning models in recommender systems is explored. By establishing the interaction between the intelligence and the environment, reinforcement learning can optimise recommendation strategies through trial and error and reward. This approach can adapt and improve recommendation results in dynamically changing environments. Finally, some challenges and future research directions for machine learning algorithms in recommender systems are presented. These include aspects such as data sparsity, the cold-start

problem, and the balance between personalisation and privacy. Future research could focus on exploring more complex models, improved optimisation algorithms and finer user feedback mechanisms to further enhance the performance of recommender systems. In conclusion, this paper provides a comprehensive discussion on the application of machine learning algorithms in the development of recommender systems. By introducing new techniques and improving traditional methods, the accuracy and personalisation of recommender systems could be improved and provide users with a better recommendation experience.

References

- [1] Zhou, W., Cao, D., Xu, Y., Liu, B., (2020) A review of recommender system research. *Journal of Hebei University of Science and Technology*, 41(01), 76-87.
- [2] Wei, S., Ye, N., Zhang, S., Huang, X., & Zhu, J. (2012). Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure. In 2012 international conference on computer science and service system, 2038-2041.
- [3] Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1), 143-177.
- [4] Cui, Z., Xu, X., Fei, X. U. E., Cai, X., Cao, Y., Zhang, W., & Chen, J. (2020). Personalized recommendation system based on collaborative filtering for IoT scenarios. *IEEE Transactions on Services Computing*, 13(4), 685-695.
- [5] Wei, H., Zhang, W., (2021) A collaborative filtering recommendation algorithm with improved user similarity[J]. *Electronic Design Engineering*, 29(17), 30-34.
- [6] He, X., He, F., Xu, L., et al. (2022) Determination of optimal number of clusters for K-Means algorithm. *Journal of University of Electronic Science and Technology*, 51(06), 904-912.
- [7] Lin, Q., (2023) A gene clustering algorithm for CCA-hierarchical clustering. *Journal of Harbin Institute of Technology*, 1-6.
- [8] Sun, H., Han, Z., (2018) A collaborative filtering improvement algorithm incorporating item popularity factor. *Small microcomputer systems*, 39(4), 638-643.
- [9] Min, L., Chunyan, W., & Yuguang, Y. (2010). The research of FP-growth method based on Apriori Algorithm in MDSS. In 2010 International Conference on Digital Manufacturing & Automation, 2, 770-773.
- [10] Luy, L., Jiang, H., (2022) Research on big data clustering method based on improved artificial neural network. *Journal of Anyang Normal College*, 2022(05), 20-23.
- [11] Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 565-573.
- [12] Kang, W. C., & McAuley, J. (2018). Self-attentive sequential recommendation. In 2018 IEEE international conference on data mining, 197-206.
- [13] Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.
- [14] Liu, X., Xie, Y., (2022) A dynamic user interest recommendation algorithm based on attention mechanism. *Computer Science and Applications*, 12(2), 11.
- [15] Meliaz, M. R., Dikairono, R., Purnama, I. K. E., & Purnomo, M. H. (2023). Deep Reinforcement Learning Control Strategy at Roundabout for i-CAR Autonomous Car. In 2023 International Seminar on Intelligent Technology and Its Applications, 473-478.
- [16] Cai, J., Gou, W., Liu, Y., (2023) Research on online integral reinforcement learning algorithm based on actor-critic framework. *Journal of Electronic Measurement and Instrumentation*, 37(03), 194-201.
- [17] Yao, Y., Ji, M. Yan., X. (2023) Research on automatic driving technology based on improved PPO algorithm. *Electronic Measurement Technology*, 46(08), 162-168.

- [18] Hu, Z., (2023) Reconsideration of scientific and technological achievements evaluation under the perspective of recommender system. Science and Science and Technology Management, 44(08), 66-80.