

Retraction

Retracted: Personalized Movie Recommendation Method Based on Deep Learning

Mathematical Problems in Engineering

Received 25 July 2023; Accepted 25 July 2023; Published 26 July 2023

Copyright © 2023 Mathematical Problems in Engineering. 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.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] J. Liu, W. Choi, and J. Liu, "Personalized Movie Recommendation Method Based on Deep Learning," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6694237, 12 pages, 2021.

Research Article

Personalized Movie Recommendation Method Based on Deep Learning

Jingdong Liu ¹, Won-Ho Choi,² and Jun Liu³

¹China and South Korea Institute of New Media, Zhongnan University of Economics and Law, Wuhan 430073, Hubei, China

²Division of Digital Contents Dongseo University, 47, Jurye-ro, Sasang-gu, Busan 617-716, Republic of Korea

³Academy of Arts and Media, China University of Geosciences, Wuhan 430074, Hubei, China

Correspondence should be addressed to Jingdong Liu; liujingdong@zuel.edu.cn

Received 15 December 2020; Revised 15 January 2021; Accepted 1 February 2021; Published 19 February 2021

Academic Editor: Sang-Bing Tsai

Copyright © 2021 Jingdong Liu 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.

With the rapid development of network technology and entertainment creation, the types of movies have become more and more diverse, which makes users wonder how to choose the type of movies. In order to improve the selection efficiency, recommend Algorithm came into being. Deep learning is a research field that has received extensive attention from scholars in recent years. Due to the characteristics of its deep architecture, deep learning models can learn more complex structures. Therefore, deep learning algorithms in speech recognition, machine translation, image recognition, and other fields have achieved impressive results. This article mainly introduces the research of personalized movie recommendation methods based on deep learning and intends to provide ideas and directions for the research of personalized movie recommendation under deep learning. This paper proposes a research method of personalized movie recommendation methods based on deep learning, including an overview of personalized recommendation and collaborative filtering recommendation algorithms, which are used to conduct research experiments on personalized movie recommendation methods based on deep learning. The experimental results in this paper show that the accuracy of the training set of the Seq2Seq model based on the LSTM recurrent neural network reaches 96.27% and the accuracy of the test set reaches 95.89%, which can be better for personalized movie recommendation.

1. Introduction

In recent years, with the improvement of people's living standards and the rapid spread of mobile Internet, more and more information is flooding the Internet [1]. Because different users have different hobbies, areas of interest, personal experience, etc., it is difficult for users to filter the information they are interested in from the massive information. How to use the big data that has emerged with the rise of mobile Internet and social media to serve users and carry out personality Chemical recommendation has become the focus of research [2]. Recommendation algorithm is a type of machine learning algorithm that is very closely related to real life. It refers to a type of algorithm that does not require users to provide clear needs but models users' interests by analyzing their historical behaviors, so as to actively recommend products to users that can meet their

interests and needs. Among the various recommendation algorithms, the collaborative filtering algorithm is the most widely used and representative algorithm. The basic idea is to use the preferences of a group with similar interests and common experience to recommend what users are interested in. Individuals give a considerable degree of response (such as scoring) to the information through a cooperative mechanism and record it to achieve the purpose of filtering and help users filter information [3]. The response is not necessarily limited to those of particular interest, and the record of particularly uninteresting information is also very important.

Traditional user interest modeling methods are difficult to express the essential information of the data and require manual extraction of features, so that researchers need to spend a lot of time and energy on data labeling, processing, and feature extraction, and different data need to be

different. Feature extraction and the extracted features are not necessarily effective. The effect of feature extraction often determines the performance of the algorithm. In recent years, deep learning has been favored by researchers [4]. It can accurately express more necessary data information through multilayer nonlinear computing units and effectively reduce the difficulty of model training through unsupervised learning.

Chen found that classification is one of the most popular topics in hyperspectral remote sensing [5]. In the past two decades, experts have proposed many methods to deal with the classification of hyperspectral data, but most of them did not extract hierarchically. Chen introduced the concept of deep learning to the classification of hyperspectral data for the first time, first by following the classification based on classical spectral information to verify the qualifications of stacked autoencoders; secondly, they proposed a new method of spatially dominant information classification; A novel deep learning framework is proposed to integrate these two functions, from which the highest classification accuracy can be obtained. The framework is a mixture of principal component analysis, deep learning architecture, and logistic regression. Specifically, as a deep learning architecture, stacked autoencoders are designed to obtain useful advanced features. This research lacks experimental data support [6]. Alhamid believes that context-aware recommendation offers the potential to use social content and use relevant tags and rating information to personalize content searches in a given context. Recommendation systems solve the problem of trying to identify relevant resources from a large number of online available choices. As a result, Alhamid proposed a new recommendation model that can personalize recommendations and improve the user experience by analyzing the context when users wish to access multimedia content; empirical analysis is performed on the data set to prove use of potential preferences to rank items in a given context; use optimization functions to maximize the average accuracy of result recommendations. This method is not very innovative [7]. Barman found that search engines have become an indispensable part of people's daily lives. They can help users find specific information from a large amount of data stored on the Internet. The query recommendation function of search engines can respond to users' original queries and provide users with Recommend multiple alternative queries. For different users from a specific geographic area, these suggestions remain basically the same, but the acceptability of these alternative queries often varies from person to person [8]. In this work, Barman D proposed a personalized recommendation system based on genetic algorithms [9] and used the search logs of commercial search engines to evaluate the proposed method. This research is not practical and not suitable for popularization in practice [10].

The innovations of this paper are as follows: (1) propose a collaborative filtering personalized movie recommendation algorithm; (2) construct a user interest model based on Seq2Seq; (3) design a personalized movie recommendation system based on deep learning.

2. Methods of Personalized Movie Recommendation Methods Based on Deep Learning

2.1. Personalized Recommendation. Through personalized recommendation by extracting the user's historical information features, it is convenient and accurate for the user to mine the things he may like from the large database and make personalized recommendation for each user [11]. For example, in the e-commerce market [12], personalized recommendation algorithms can stimulate users' potential purchase desires and reduce the time for users to select products, thereby facilitating users' shopping methods; in the news or video fields, personalized recommendation algorithms can provide users with recommendations. The information of his "appetite" improves the user's reading efficiency, reduces the time for users to select product content, and can also attract users' interest in the product [13]. Nowadays, recommended websites can obtain user behaviors such as length of stay, favorite links, and number of likes. These behaviors are roughly divided into explicit feedback behaviors and implicit feedback behaviors. Explicit feedback behaviors can directly present user preferences. Commonly it is the user's rating of the item [14].

The implicit feedback behavior cannot clarify the user's preferences [15]. A common implicit feedback behavior is the user's web browsing record. The user may not be interested in the item when browsing the web but may click and browse the item out of curiosity or unintentionally [16]. Although the record does not clearly know the user's preferences, the implicit data obtained by the general website account for a large proportion. Therefore, based on the use of explicit feedback data, it is necessary to dig out the implicit feedback data value meaning content, so as to achieve personalized recommendation [17].

According to the different data types, the algorithms needed for personalized recommendation are mainly composed of two types: the first is a recommendation algorithm based on content, and the second is a recommendation algorithm based on collaborative filtering. The main idea of the content-based recommendation algorithm is to recommend to users the information with the greatest similarity in the content of the items they like and the information they have followed; the personalized recommendation system basically uses a collaborative filtering algorithm [18], and its core recommendation idea is as follows: the user has other users with similar preferences and then recommends to the user items that other users have purchased but this user has not purchased [19].

2.2. Collaborative Filtering Recommendation Algorithm

2.2.1. Similarity Calculation. The measure of similarity between users is generally compared by the method of vector calculation. Among these comparison methods, Pearson similarity and cosine similarity are the most commonly used [20].

The calculation formula of cosine similarity is as follows, where $\text{sim}(x, y)$ represents the similarity between users x and y and $N(x)$ and $N(y)$ represent the collection of ratings of users x and y , respectively:

$$\text{sim}(x, y) = \frac{|N(x) \times N(y)|}{\sqrt{|N(x)||N(y)|}} \quad (1)$$

The calculation formula of Pearson's correlation coefficient is as follows:

$$\text{sim}(x, y) = \frac{\sum_{i \in N(x) \cap N(y)} (R_{xi} - \bar{R}_x) * (R_{yi} - \bar{R}_y)}{\sqrt{\sum_{i \in N(x) \cap N(y)} (R_{xi} - \bar{R}_x)^2} * \sqrt{\sum_{i \in N(x) \cap N(y)} (R_{yi} - \bar{R}_y)^2}} \quad (2)$$

When recommending the target user x , similarity calculation is also needed to find the n movies most similar to movie m [21]. Here, it is represented by the set $S(m, n)$; the user x 's interest in movie m can be calculated by the following formula:

$$X(x, n) = \sum_{j \in S(m, n) \cap N(x)} \text{sim}(n, j) p_{xj}. \quad (3)$$

where $N(x)$ represents the set of movies that user x likes and p_{xj} represents the degree of user x 's preference for movie j .

2.2.2. Matrix Decomposition Recommendation Algorithm. The basic principle of the matrix decomposition recommendation algorithm is as follows: analyze the user's historical information and construct a rating matrix, which is composed of user ID, item ID, and rating [22]. Then, the user item rating matrix is decomposed into two low-dimensional feature matrices through matrix decomposition algorithm. Finally, these two low-dimensional matrices are used to estimate items that users have not commented on [23]. The initial user-item rating matrix $R^{n \times m}$ is decomposed, and the user vector representation and the item vector representation are obtained, respectively, and then two feature matrices are obtained [24], namely, user feature matrix $U^{k \times n}$ and item feature matrix $V^{k \times m}$, where n is the number of users, m is the number of items, and k is the dimension of the hidden vector feature space [25, 26]. Multiplying the two matrices obtained by decomposition is the predicted score:

$$R^* = U^T V. \quad (4)$$

The algorithm trains the model by optimizing the difference between the predicted score R and the real score R^* , that is, optimizing the objective function:

$$E = I(R - U^T V)^2 = \sum_{i=1}^N \sum_{j=1}^M I_{ij} (R_{ij} - U_i^T V_j)^2. \quad (5)$$

To prevent overfitting, redefine the objective function:

$$E = \sum_{i=1}^N \sum_{j=1}^M (R_{ij} - U_i^T V_j)^2 + \lambda (\|U_i\|^2 + \|V_j\|^2). \quad (6)$$

Use the stochastic gradient descent method to solve the matrices U and V [27]. Then, the partial derivatives of E with respect to U and V are expressed as the following formulas:

$$\frac{\partial E}{\partial U} = -2V + 2\lambda U, \quad (7)$$

$$\frac{\partial E}{\partial V} = -2U + 2\lambda V.$$

Get updated U and V , where α is the learning rate:

$$\begin{aligned} U &= U + \alpha(V - \lambda U), \\ V &= V + \alpha(U - \lambda V). \end{aligned} \quad (8)$$

2.2.3. Probability Matrix Factorization. Probabilistic matrix decomposition is to introduce a probability model on the basis of matrix decomposition to optimize. The introduction of the probability model has greatly improved the performance of matrix factorization and further improved the accuracy of the matrix factorization model [28]. Probability matrix decomposition has two leading assumptions: one is that the difference between the overall rating matrix R of the user and the inner product R of the eigenvectors of the user and the movie obeys the Gaussian distribution of variance [29]; the second is that the eigenvector matrix U of the user and the movie's elements of the eigenvector matrix V , respectively, obey the Gaussian distribution with the mean value being 0 and the variance being ϕ_u and ϕ_v [30,31].

According to hypothesis one, the probability density function of R can be obtained, where I_{ij} is an indicative function representing whether user i has made an evaluation for movie j ; if the evaluation has been made, its value is 1; otherwise, it is 0:

$$p(R|U, V, \phi^2) = \prod_{i=1}^N \prod_{j=1}^M \left[N\left(\frac{R_{ij}}{U_i^T V_j}, \phi^2\right) \right]^{I_{ij}}. \quad (9)$$

According to hypothesis two, U and V probability density functions can be obtained:

$$p(U, \phi_u^2) = \prod_{i=1}^N N\left(\frac{U_i}{0}\right), \quad \phi_u I, \quad p(V, \phi_v^2) = \prod_{i=1}^M N\left(\frac{V_j}{0}\right), \quad \phi_v I. \quad (10)$$

The stochastic gradient descent method is used to solve the matrices U and V [32]. Then, the partial derivatives of E with respect to U and V are expressed as the following formulas:

$$\begin{aligned}\frac{\partial E}{\partial U} &= -V + \lambda_u U, \\ \frac{\partial E}{\partial V} &= -U + \lambda_v V.\end{aligned}\quad (11)$$

Get updated U and V , where α is the learning rate:

$$\begin{aligned}U &= U + \alpha(V - \lambda_u U), \\ V &= V + \alpha(U - \lambda_v V).\end{aligned}\quad (12)$$

The method part of this article uses the above method for the research of personalized movie recommendation algorithm based on deep learning. The specific process is shown in Figure 1.

3. Experiment on Personalized Movie Recommendation Method Based on Deep Learning

3.1. Based on Seq2seq User Interest Model Construction. The personalization of personalized search is reflected in the different search results obtained by different users. First, obtain user interests through related algorithms and express and describe the user's personalized information. Secondly, the personalized information is integrated into the search algorithm or related operations such as expanding the query sentence during the query process and dynamically adjusting the search results [33]. The Seq2Seq model in deep learning is used to characterize the user's interest, and then it is integrated into the sorting algorithm of the search engine. The coding end of the Seq2Seq model is based on the LSTM recurrent neural network, which has a longer memory capacity than RNN. The decoder also uses LSTM recurrent neural network [34].

First, classify the movie titles according to the authors, put together the movie titles of the same director, and construct them into short texts. All authors in the data set are made into short text data sets; secondly, the data are input into the Seq2Seq model, and the input and output are both It is the same data for training until the loss converges and stabilizes. This process is to adjust the weight parameters of the neural network through training, and let the model capture the input semantic information; again input the relevant short text of each director into the trained code In the device, the output result at the last time step is the interest feature vector of the current audience, and it is saved for subsequent use [35]. Finally, each movie title of the search result must be input into the encoder, the generated vector and the interest vector of the audience currently using the system are calculated for similarity, and the calculation result is merged into the Lucene sorting algorithm, and then the search result is reproduced. Sort to achieve personalized search results [36].

3.1.1. Model Structure. This article uses the Seq2Seq model to express user interests. The whole model constructed in this paper is mainly divided into four levels. The first level is the data embedding layer, which embeds the text data in a

vectorized manner. This paper chooses the word embedding method instead of the one-hot method. The second layer is the coding layer, which is based on the LSTM neural network and encodes text data. The third layer is the attention layer. Through the attention layer, the decoder can use a different intermediate vector at each time step. The fourth layer is the decoding layer, and the decoding layer also uses LSTM recurrent neural network to decode and output according to the intermediate vector [37, 38].

3.1.2. Data Preprocessing Vocabulary Construction. The original data need to be processed by removing the stop words, and the stop word list used is the stop word list for English text processing in the natural language processing toolkit (NLTK) related to python. Before the vocabulary is built, punctuation marks in the text are eliminated, which helps to reduce the number of unregistered words. In the process of constructing the vocabulary, the case of words is distinguished, which helps to understand semantics and enhance the expression effect. Construct the vocabulary according to the word frequency and arrange it according to the size of the word frequency. After that, the word is represented by a unique ID, that is, its sorted position in the word frequency database. The word ID is built into a batch and sent to the network, and the network performs word embedding into embedded computing that it is used for subsequent network training [39].

3.1.3. Embedding Processing Layer. The Embedding layer uses the training skip-gram model to construct word vectors. First, the movie title corpus data are counted on word frequency and sorted according to word frequency, and a fixed position is determined for each word. Next, perform one-hot encoding. Finally, a word vector matrix is generated through neural network training, where the row vector corresponding to each position is the distributed word vector of the current word [40].

3.1.4. Attention Layer Construction. It is the introduction of the attention mechanism into the model. The basic idea of the attention mechanism is to break the limitation that the traditional encoder-decoder depends on a fixed vector output at the last time step of the encoding stage when decoding. Attention mechanism is realized by saving the intermediate output results generated by the encoder according to the input sequence and then training a model to selectively learn these inputs and associate the output sequence with it when the model is output [41].

3.2. Design a Personalized Movie Recommendation System Based on Deep Learning

3.2.1. Demand Analysis

(1) User Needs. The functions that users can use generally include account management, movie search, user ratings, user reviews, personalized movie recommendations, movie

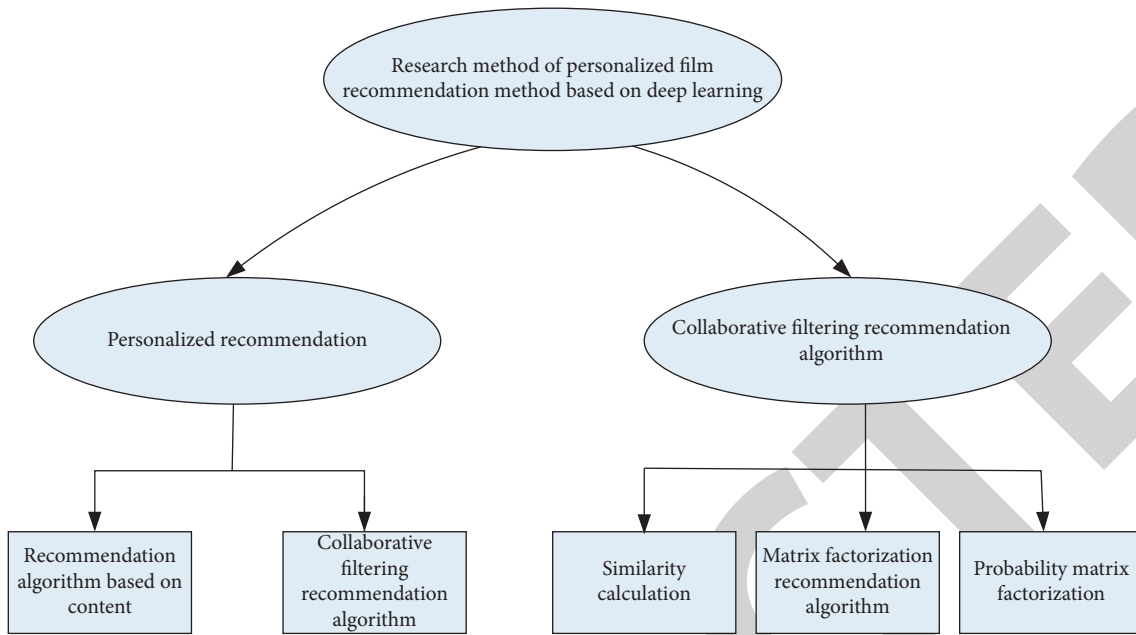


FIGURE 1: Part of the technical process of this method.

nonpersonalized recommendations, and user information management. Account management includes account login, account logout, and account registration; movie search includes movie search and movie playback sources; user information management includes user nickname modification and user password modification.

(2) *Administrator Requirements.* In addition to the user rights, the system administrator needs to undertake the operation and maintenance tasks of the entire movie recommendation system to ensure the normal and stable operation of the system. These include user information management, website traffic statistics, and website information maintenance. User information management includes banning users and modifying user information and website traffic statistics; website information maintenance includes adding movies, deleting movies, updating movie information, deleting comments, and viewing comments.

(3) *Performance Requirements.* Ensuring system performance, high availability, and scalability.

3.2.2. System Structure. This system adopts the B/S architecture, uses the MVC development model as the system's microsolution, and uses the microservice architecture as the system's macrosolution to design and develop a large-scale Web system. The development of the whole system takes MVC development model as the development model for designing and creating Web applications and chooses SpringMVC as the framework of MVC development model for development.

The interface layer is dominated by html pages, which are implemented through technologies such as JavaScript, SpringMVC, and Json. Among them, it mainly communicates with the server through the Http network

communication protocol, and the users of the system interact with the system by browsing the web.

The business logic layer is mainly implemented by SpringMVC, based on the controller in the MVC development model, connecting the interface layer and the data access layer. Through the business logic layer, the data access layer can write data to the interface layer, and the data access layer can also send data back to the interface layer.

The data access layer mainly uses the Hibernate framework to complete object-relational mapping. First, it encapsulates JDBC lightly and establishes a mapping relationship between the model in the MVC development model and the database table to complete data persistence and other operations.

3.2.3. Database Design

Relational Database. The recommendation system needs a lot of data to support when updating the recommendation model, and there is a certain relationship between data and data, and the relational database is based on the relational model, so the commonly used relational database Mysql is used Store system information.

(1) *Nonrelational Database.* This type of data requires a performance-oriented database system. Redis is a non-relational database, which is based on memory key-value storage, which is more efficient in storage. For example, after logging in to the movie recommendation system, I regret storing the logged-in user's Session information in Redis. The storage form of the user's Session information in Redis is as follows: Key = User ID; Value = Session Information.

3.2.4. Core Module Design

(1) *Login Module.* This module is mainly responsible for the user's identity verification when using the system and grants different module access permissions based on the user's identity and the access control list provided by Spring Security.

(2) *Movie search Module.* This module is responsible for searching out the movie corresponding to the user when the user searches for a movie and providing the playback source of the movie to the user.

(3) *Offline Calculation Module.* By clustering the user's ratings of all movies and the category of the movie, a part of the movies that users are most likely to be interested in are selected as a list of candidate movies, and then a deep network structure model is used to the candidate movie list to recommend and sort, which is based on deep learning.

(4) *Online Recommendation Module.* Sort and recommend the movie candidate list obtained by the offline module by using an improved movie recommendation algorithm based on deep learning.

(5) *Information Management Module.* The system administrator can add, delete, and update information on the movie resources of the system, check the comments of each movie, and delete false comments.

(6) *Website Traffic Statistics.* It is responsible for statistics of various traffic on the website, which is convenient for administrators to do operation and maintenance management.

(7) *User Information Feedback Module.* Support users to give feedback on recommended movie information in the system, including user ratings and user comments.

This experiment part proposes the above steps to be used in the research experiment of personalized movie recommendation method based on deep learning. The specific process is shown in Table 1.

4. Personalized Movie Recommendation Method Based on Deep Learning

4.1. *Development Status of Intelligent Recommendation System.* Nowadays, the recommendation system as a kind of intelligent information service has been applied in all walks of life, for example, the well-known e-commerce product recommendation, hot topic recommendation of news websites, high-quality content recommendation of video and music, keyword recommendation of search engines, and recommendation of similar documents in academic websites. This article counts the well-known apps on the market that contain smart recommendations, as shown in Table 2.

The recommendation system uses the user's historical behavior information to infer the user's interests and preferences, so as to provide recommendations for them. The personalized recommendation system can not only

make recommendations based on users' preferences but can also recommend new items that users might like when the users are not clear about their preferences. This is called the novelty of recommendation.

4.2. Personalized Movie Recommendation User Analysis

- (1) Count the average usage time of the application platform with movie personalized recommendation function in the daily life of each user layer, and draw it into a chart, as shown in Table 3 and Figure 2.

It can be seen from the chart that Saturday and Sunday are the two days when users spend the most time online. The platform can set the number of personalized recommendations on Saturdays and Sundays to be more frequent, and there are more types of pushes; users online use time on weekdays after Monday. Relatively short, the platform can be set to accurately push according to users' likes.

- (2) Calculate the time period that users use the application platform with the personalized recommendation function of movies and draw them into charts, as shown in Table 4 and Figure 3.

It can be seen from the graph that users often use application platforms with personalized movie recommendation functions during commuting hours, lunch time, dinner time, etc. It should be adapted to the current conditions, and more personalized movie recommendations for users during these time periods will surely achieve a better recommendation effect.

4.3. Experimental Results

- (1) This paper calculates the similarity algorithm for the collaborative filtering algorithm based on personalized recommendation and the Seq2Seq model, fuses the similarity through the fusion algorithm, and finally completes the movie recommendation. The recommendation is the Top-N method. Analyze and compare relevant experimental results. Through the fusion algorithm, the similarity between the movie similarity obtained by the item-based collaborative filtering recommendation algorithm (Item-CF) and the movie based on the Seq2Seq model is fused and statistically sorted out. Draw a chart, as shown in Table 5 and Figure 4.

It can be seen from the chart that the similar fusion ratio of Item-CF and Seq2Seq is 1:0, which means the index result is obtained by the Item-CF algorithm. It can be seen from various evaluation indicators that when the Seq2Seq fusion ratio is relatively high, the coverage rate is significantly higher than that of the traditional collaborative filtering algorithm. As the proportion of Seq2Seq fusion decreases, the accuracy, recall, and novelty are all improved, while the coverage rate gradually decreases.

TABLE 1: Experimental steps in this article.

Research experiment on personalized movie recommendation method based on deep learning	3.1. Based on Seq2Seq user interest model construction	1. Model structure 2. Data preprocessing vocabulary construction 3. Embedding processing layer 4. Attention layer construction
	3.2. Design a personalized movie recommendation system based on deep learning	1. Demand analysis 2. System structure 3. Database design 4. Core module design

TABLE 2: App with smart recommendations.

Application field	App	Uses
E-commerce	Taobao, Jingdong, Buy together, Amazon	Online shopping
Film and television	Youtube, Youku, Tencent video, Bilibili, Aiqiyi, Douban	Watching TV and movies
Music software	NetEase Cloud Music, Kugou music, Kuwo music, QQ music, Xiami Music	Listen to the music
Radio station	Lychee fm, Radio Himalayan	Listen to the radio
News and information	Toutiao, Zite	News browsing
Life service	Meituan, Eleme	Eating, lodging, traveling
Social contact	QQ, Weibo, Wechat	Exchange, make friends
Search engines	Baidu, Google	Search for things you want to know

TABLE 3: Average user usage time (unit: minute).

User	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Primary and secondary school students	112.3	106.4	101.2	110.1	116.4	134.8	141.2
College students	164.1	171.3	164.6	175.3	154.9	212.3	230.6
Office worker	108.6	92.7	95.6	100.5	124.6	176.7	200.4

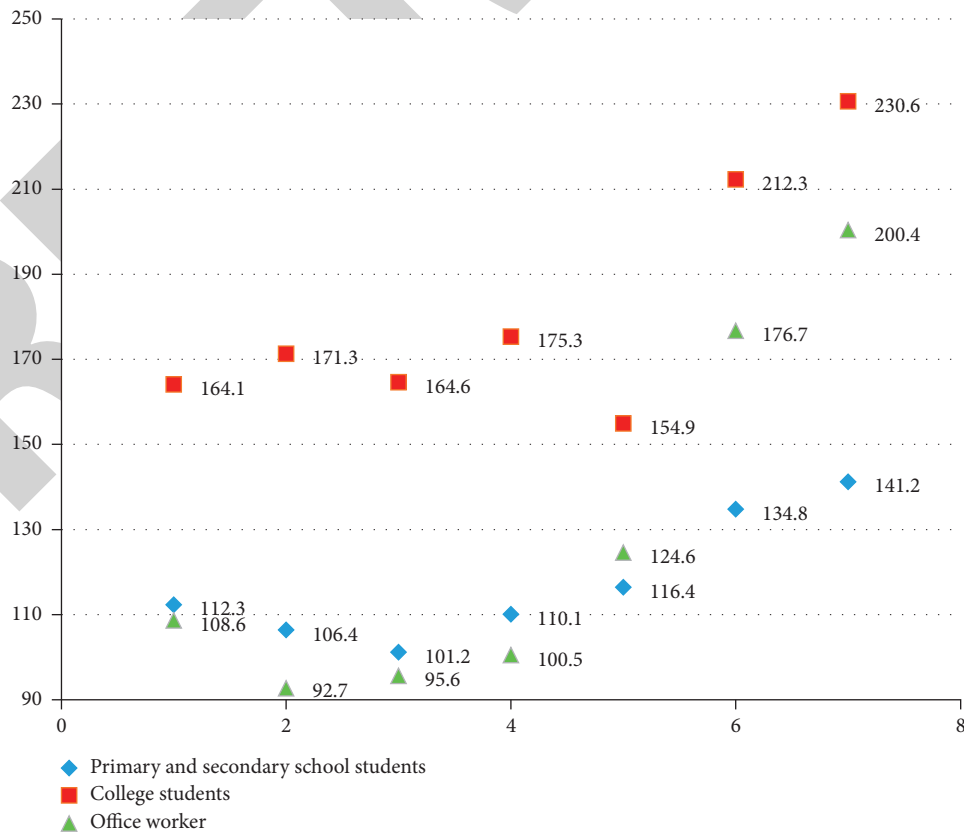


FIGURE 2: Average user usage time (unit: minute).

TABLE 4: User viewing time period.

Period	Percentage (%)
0:00–6:00	3.62
6:00–9:00	6.23
9:00–12:00	13.91
12:00–15:00	20.34
15:00–18:00	19.12
18:00–21:00	23.67
21:00–24:00	13.11

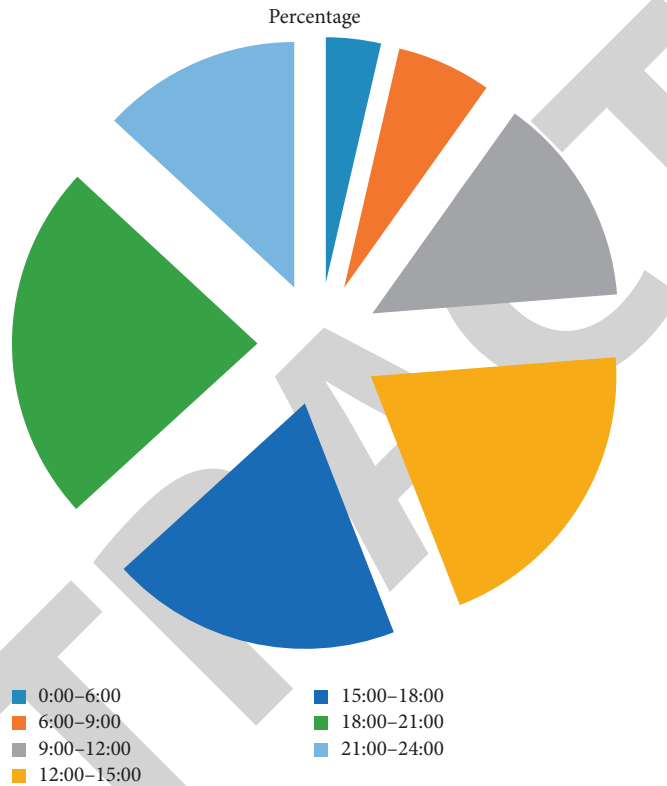


FIGURE 3: User viewing time period.

TABLE 5: Item-CF and Seq2Seq fusion results in different proportions.

Number	Similar fusion ratio of Item-CF and Seq2seq	Accuracy (%)	Recall rate (%)	Coverage (%)	Novelty
1	1:0	39.81	4.21	29.61	4.31
2	1:1	42.31	4.67	32.33	4.26
3	1:2	37.42	3.94	36.07	4.54
4	7:1	35.67	4.56	41.20	4.62
5	14:1	42.33	5.08	39.26	4.70
6	19:1	46.24	4.78	36.44	3.96
7	20:6	48.09	5.12	32.52	4.33
8	48:1	39.21	4.83	29.07	4.27
9	50:4	36.52	3.64	26.43	4.39
10	67:2	37.43	3.27	23.67	4.51
11	92:1	41.66	2.98	29.11	4.63
12	100:1	49.57	2.51	30.27	4.71

(2) The data set used in the experiment is the MovieLens data set collected by the GroupLens team of the University of Minnesota in the United States. It records users' ratings of movies and is a standard data set for

personalized recommendation algorithm evaluation. The MovieLens dataset used in this article includes MovieLens-100k dataset and MovieLens-1M dataset. Based on the MovieLens-100k data set and MovieLens-

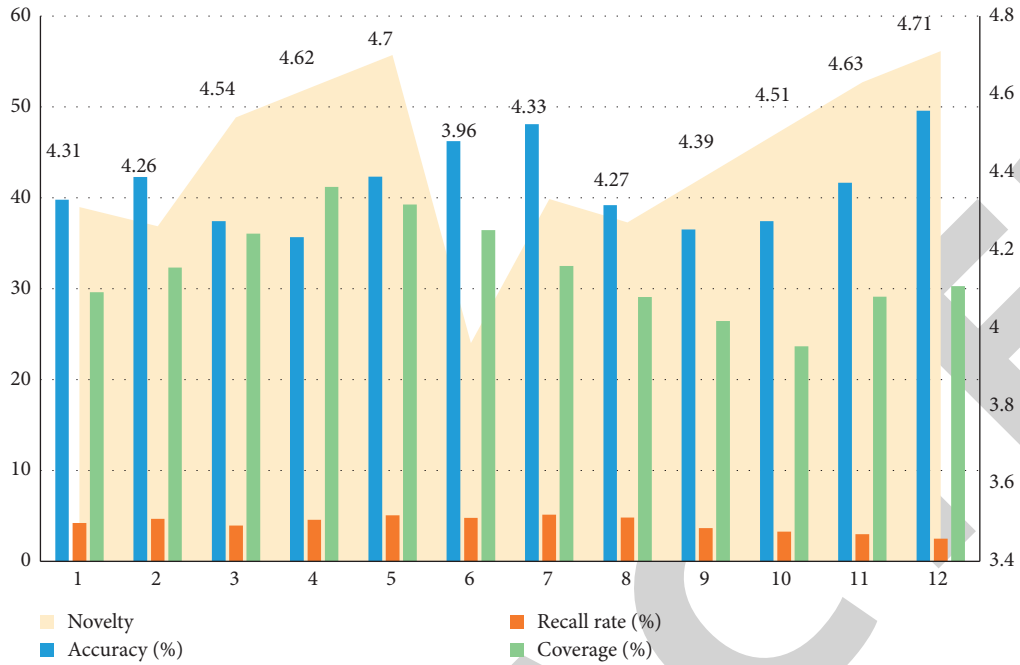


FIGURE 4: Item-based collaborative filtering recommendation algorithm and Seq2Seq fusion results in different proportions.

TABLE 6: Comparative experimental results.

User activity	MovieLens-100k			MovieLens-1M		
	HR	NDCG	ILS	HR	NDCG	ILS
0.2	0.6481	0.3421	0.4568	0.7126	0.5027	0.4613
0.3	0.6527	0.3519	0.4631	0.7217	0.5112	0.4717
0.4	0.6582	0.3607	0.4724	0.7305	0.5204	0.4806
0.5	0.6643	0.3687	0.4811	0.7396	0.5293	0.4912
0.6	0.6709	0.3724	0.4901	0.7421	0.5348	0.5071

1M data set, starting from three aspects, the output size of the hidden layer is 8. Regarding the models GMF, MLP, NeuMF, and Seq2Seq, the accuracy and diversity of the two data sets are compared Experiment to verify the superiority of the Seq2Seq model is proposed in this chapter. The specific results are shown in Table 6 and Figure 5.

It can be seen from the graph that, as the user activity increases, the accuracy fluctuates slightly and the diversity rises slightly; the increase in activity indicates that the proportion of user activity obtained by clustering gradually increases, compared to pure. The diversity characteristics of the activity derived from the number of times the project is evaluated are more obvious and bring more gains.

- (3) A three-layer LSTM network model is built, the number of LSTM neurons is set to 131, the learning is 0.002, the dropout value is 0.2, the number of training rounds Epoch is 27, the batch size Batch_size is 35, Adam is the model optimizer, and Sigmoid function is used as the activation function. When the model is stable, the setting is that the LSTM layer contains 131 neurons and the batch_size of each batch of data is 70.

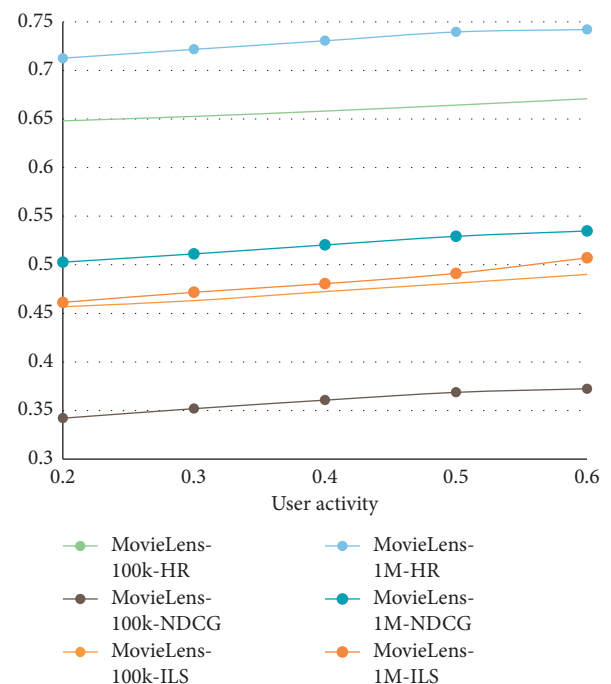


FIGURE 5: Comparison of experimental results.

TABLE 7: Loss function and accuracy rate changes.

Epoch	Loss training	Loss test	Accuracy training (%)	Accuracy test (%)
0~3	0.0152	0.0146	71.24	64.14
4~6	0.0131	0.0137	79.56	75.63
7~9	0.0127	0.0121	82.31	80.41
10~12	0.0114	0.0117	85.47	82.67
13~15	0.0097	0.0106	90.26	89.17
16~18	0.0086	0.0095	92.51	93.67
19~21	0.0082	0.0089	93.38	94.59
22~24	0.0078	0.0074	95.46	95.21
25~27	0.0073	0.0069	96.27	95.89

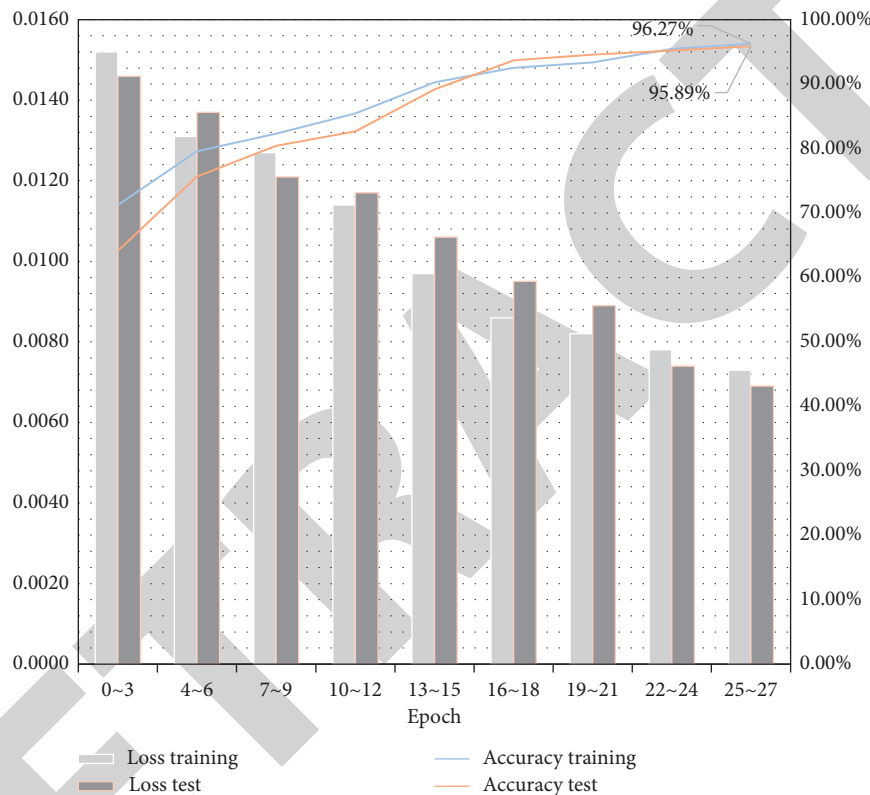


FIGURE 6: Loss function and accuracy rate changes.

After 27 iterations, the model achieves better results. The dropout value is 0.23, and Adam as a model optimizer, the sigmoid function is used as the final activation function of the model. The specific conditions of the model's loss function and accuracy rate change trend on the test set and training set are shown in Table 7 and Figure 6.

It can be seen from the graph that the loss function and accuracy of the model gradually stabilized after the epoch reached 15, and the loss function remained around 0.0080. The accuracy of the final model training set reached 96.27%, and the accuracy rate of the test set reached 95.89%. Convergence speed and model accuracy are better than traditional RNN models. Compared with traditional RNN models, processing performance has been greatly improved.

5. Conclusions

In recent years, with the rapid development of information technology and the Internet, watching movies through the Internet has become a habit of many people. However, because people cannot quickly obtain their favorite movie content from the massive movie resources, this makes the overload of movie information more and more serious. As one of the important means to alleviate the problem of information overload, the recommendation system can help users quickly find favorite movie content and bring users a good experience. Therefore, it is widely used in well-known movie and video websites at home and abroad and has great commercial value.

The advent of the 5G era has put forward higher requirements for Internet technology. As one of the important areas, the recommendation system has been deeply applied to all aspects of the Internet. It helps people provide

solutions from massive information retrieval and changes the original passive search method. To take the initiative and bring certain economic benefits to related enterprises, it has been widely recognized by the industry. At the same time, deep learning technology has shined in the field of classification and prediction in recent years, but in the field of recommendation, the application of this technology is still in a period of rapid growth. How to combine the two more effectively is an important direction of current research.

This article begins by focusing on the research background and significance of the thesis. Through the research and judgment of the current research status in this field, combined with the development and application of algorithm technology, it analyzes and summarizes the common problems in the algorithm. Then, the related theories and technologies needed in this article are researched and organized, several basic algorithms of recommendation systems and the current popular social network-based recommendation algorithms are theoretically studied, and then the foundation of deep learning algorithms is explained. This article has conducted certain research on the movie recommendation system and its combination with deep learning, but there are still many shortcomings, which require continuous learning and hard work. The data set used in the model test in this paper is very small compared with the calculation required for actual application, and there is still a certain gap in the amount of data.

Data Availability

No data were used to support this study.

Disclosure

The content of the manuscript has not been published or submitted for publication elsewhere.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

All the authors have seen the manuscript and approved the submission of the article.

References

- [1] S. Wan, X. Li, Y. Xue, W. Lin, and X. Xu, "Efficient computation offloading for Internet of Vehicles in edge computing-assisted 5G networks," *The Journal of Supercomputing*, vol. 76, pp. 2518–2547, 2020.
- [2] L. M. T. Pham, L. T. T. Tran, P. Thipwong, and W. T. Huang, "Dynamic capability and organizational performance," *Journal of Organizational and End User Computing*, vol. 31, no. 2, pp. 1–21, 2019.
- [3] X. Li, Y. Zhu, and J. Wang, "Highly efficient privacy preserving location-based services with enhanced one-round blind filter," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, p. 1, 2019.
- [4] L. Wu, Q. Zhang, C.-H. Chen, K. Guo, and D. Wang, "Deep learning techniques for community detection in social networks," *IEEE Access*, vol. 8, pp. 96016–96026, 2020.
- [5] G. Wang, Y. Yao, Z. Chen, and P. Hu, "Thermodynamic and optical analyses of a hybrid solar CPV/T system with high solar concentrating uniformity based on spectral beam splitting technology," *Energy*, vol. 166, pp. 256–266, 2019.
- [6] Y. Chen, Z. Lin, X. Zhao et al., "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2017.
- [7] M. F. Alhamid, M. Rawashdeh, H. Dong et al., "Exploring latent preferences for context-aware personalized recommendation systems," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 1–9, 2017.
- [8] Z. Lv, X. Li, and W. Li, "Virtual reality geographical interactive scene semantics research for immersive geography learning," *Neurocomputing*, vol. 254, pp. 71–78, 2017.
- [9] N. Metawa, M. Elhoseny, M. Kabir Hassan, and A. E. Hassanien, "Loan portfolio optimization using genetic algorithm: a case of credit constraints," in *Proceedings of 12th International Computer Engineering Conference, ICENCO*, pp. 59–64, Cairo, Egypt, December 2016.
- [10] D. Barman, R. Sarkar, A. Tudu, and N. Chowdhury, "Personalized query recommendation system: a genetic algorithm approach," *Journal of Interdisciplinary Mathematics*, vol. 23, no. 2, pp. 523–535, 2020.
- [11] X. Chowdhury, G. Zhang, and S. Ma, "Deep learning," *International Journal of Semantic Computing*, vol. 10, no. 3, pp. 417–439, 2016.
- [12] C.-H. Wu and S.-B. Tsai, "Using DEMATEL-based ANP model to measure the successful factors of E-commerce," *Journal of Global Information Management*, vol. 26, no. 1, pp. 120–135, 2018.
- [13] G. Litjens, T. Kooi, B. E. Bejnordi et al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, no. 9, pp. 60–88, 2017.
- [14] S. Setio, P. Pastor, A. Krizhevsky et al., "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2016.
- [15] L. Fabisiak, "Web service usability analysis based on user preferences," *Journal of Organizational and End User Computing*, vol. 30, no. 4, pp. 1–13, 2018.
- [16] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.
- [17] X. Zhang and D. Wang, "Deep learning based binaural speech separation in reverberant environments," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1075–1084, 2017.
- [18] Z. Lv and L. Qiao, "Optimization of collaborative resource allocation for mobile edge computing," *Computer Communications*, vol. 161, pp. 19–27, 2020.
- [19] W. Li, H. Fu, L. Yu et al., "Stacked Autoencoder-based deep learning for remote-sensing image classification: a case study of African land-cover mapping," *International Journal of Remote Sensing*, vol. 37, no. 23-24, pp. 5632–5646, 2016.
- [20] Q.-S. Zhang and S.-C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018.

- [21] I. Y. Choi, H. S. Moon, and J. K. Kim, "Assessing personalized recommendation services using expectancy disconfirmation theory," *Asia Pacific Journal of Information Systems*, vol. 29, no. 2, pp. 203–216, 2019.
- [22] M. Kim, Y. Fang, H. Xie, J. Chong, and M. Meng, "User click prediction for personalized job recommendation," *World Wide Web*, vol. 22, no. 1, pp. 325–345, 2019.
- [23] J. Zhang, "Personalised product recommendation model based on user interest," *Computer Systems Science and Engineering*, vol. 34, no. 4, pp. 231–236, 2019.
- [24] M. Gan and R. Jiang, "FLOWER: fusing global and local associations towards personalized social recommendation," *Future Generation Computer Systems*, vol. 78, no. 1, pp. 462–473, 2017.
- [25] W. Gu, S. Dong, and M. Chen, "Personalized news recommendation based on articles chain building," *Neural Computing and Applications*, vol. 27, no. 5, pp. 1263–1272, 2016.
- [26] C. Tan, H. Li, and X. Wu, "Context-aware personalized recommendation for mobile users," *IPPTA: Quarterly Journal of Indian Pulp and Paper Technical Association*, vol. 30, no. 2, pp. 146–151, 2018.
- [27] Y. Lv, "Personalized recommendation model based on incremental learning with continuous discrete attribute optimization," *Revista de la Facultad de Ingenieria*, vol. 32, no. 2, pp. 842–849, 2017.
- [28] S. Lee and T. Ha, "Item-network-based collaborative filtering: a personalized recommendation method based on a users item network," *Information Processing & Management: Libraries and Information Retrieval Systems and Communication Networks: An International Journal*, vol. 53, no. 5, pp. 1171–1184, 2017.
- [29] G. Xiao, Q. Cheng, and C. Zhang, "Detecting travel modes using rule-based classification system and Gaussian process classifier," *IEEE Access*, vol. 7, pp. 116741–116752, 2019.
- [30] C. Shi, Z. Zhang, Y. Ji, W. Wang, P. S. Yu, and Z. Shi, "SemRec: a personalized semantic recommendation method based on weighted heterogeneous information networks," *World Wide Web*, vol. 22, no. 1, pp. 153–184, 2019.
- [31] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, "Personalized real-time movie recommendation system: practical prototype and evaluation," *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 180–191, 2020.
- [32] J. Grith, C. O. Riordan, J. Griffith et al., "Collaborative filtering," *Computer Science*, vol. 57, no. 4, p. 189, 2017.
- [33] Y. Yao, H. Tong, G. Yan et al., "Dual-regularized one-class collaborative filtering with implicit feedback," *World Wide Web*, vol. 22, no. 3, pp. 1099–1129, 2019.
- [34] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec, "Privacy-preserving distributed collaborative filtering," *Computing*, vol. 98, no. 8, pp. 827–846, 2016.
- [35] N. Polatidis and C. K. Georgiadis, "A multi-level collaborative filtering method that improves recommendations," *Expert Systems with Applications*, vol. 48, pp. 100–110, 2016.
- [36] D. Zhang, T. He, Y. Liu et al., "A carpooling recommendation system for taxicab services," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 254–266, 2017.
- [37] J. D. West, I. Wesley-Smith, and C. T. Bergstrom, "A recommendation system based on hierarchical clustering of an article-level citation network," *IEEE Transactions on Big Data*, vol. 2, no. 2, pp. 113–123, 2016.
- [38] Z. Tian, T. Jung, Y. Wang et al., "Real-time charging station recommendation system for electric-vehicle taxis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3098–3109, 2016.
- [39] H. C. Zhang and Y. L. Chang, "PKR: a personalized knowledge recommendation system for virtual research communities," *Data Processor for Better Business Education*, vol. 48, no. 1, pp. 31–41, 2016.
- [40] E. N. Cinicioglu and P. P. Shenoy, "A new heuristic for learning Bayesian networks from limited datasets: a real-time recommendation system application with RFID systems in grocery stores," *Annals of Operations Research*, vol. 244, no. 2, pp. 385–405, 2016.
- [41] A. S. Koshiyama, N. Firoozye, and P. Treleaven, "A derivatives trading recommendation system: the mid-curve calendar spread case," *Intelligent Systems in Accounting, Finance and Management*, vol. 26, no. 2, pp. 83–103, 2019.