A Deep Ranking Weighted Multihashing Recommender System for Item Recommendation

Suresh Kumar 1, Jyoti Prakash Singh 1, Vinay Kumar Jain 2, and Avinab Marahatta 3

1Department of Computer Science and Engineering, NIT Patna, India
2Department of Management (PG), MIT World Peace University, Pune, India
3Center for Multidisciplinary Studies and Innovation (CeMuSI), Kathmandu, Nepal

Correspondence should be addressed to Avinab Marahatta; avinab.marahatta@gmail.com

Received 19 July 2022; Accepted 14 September 2022; Published 10 October 2022

Copyright © 2022 Suresh Kumar 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.

Collaborative filtering (CF) techniques are used in recommender systems to provide users with specialised recommendations on social websites and in e-commerce. But they suffer from sparsity and cold start problems (CSP) and fail to interpret why they recommend a new item. A novel deep ranking weighted multihash recommender (DRWMR) system is designed to suppress sparsity and CSP. The proposed DRWMR system contains two stages: the neighbours’ formation and recommendation phases. Initially, the data is fed to the deep convolutional neural network (CNN). The significant features are extracted from CNN. The CNN contains an additional layer; the hash code is generated by minimising pairwise ranking loss and classification loss. Therefore, a weight is assigned to different hash tables and hash bits for a recommendation. Then, the similarity between users is obtained based on the weighted hammering distance; the similarity between users helps to form the neighbourhood for the active user. Finally, the rating for unknown items can be obtained by taking the weighted average rating of the neighbourhood, and a list of the top $n$ items can be produced. The effectiveness and accuracy of the proposed DRWMR system are tested on the MovieLens 100K dataset and compared with the existing methods. Based on the evaluation results, the proposed DRWMR system gives precision (0.16), the root mean squared error (RMSE) of 0.73 and the recall (0.08), the mean absolute error (MAE) of 0.57, and the $F−1$ measure (0.101).

1. Introduction

Recommendation systems (RS) have recently become common on numerous websites, recommending movies, e-commerce, music, and television programs [1]. Based on the information provided by the user, the RS recommends items for purchase [2]. Several RS have been introduced to predict the behaviour of users and provide better recommendations [3, 4]. In general, RS recommends, based on the individual interests of the user and their previous usage history, finding items with the highest preference [5]. RS is a great machine learning system to increase product sales [6, 7]. Recommendation helps the user to speed up the search process and makes it simple for them to obtain content that is interesting to them, as well as provide them with offers they would not have searched for [8, 9]. Furthermore, companies may attract customers by showing movies and TV shows relevant to their profiles [10].

The approaches to recommendation may be categorised as collaborative filtering (CF), content-based (CB), and hybrid based on the type of data gathered and how it is used in the RS [11]. CB filtering is frequently used in the RS design, which uses items’ content to select general characteristics and qualities that suit the user profiles [12]. Then, users compare items to previously liked items and recommend the best matching-items. The CF system works based on a user-item relationship. Hybrid systems utilise both forms of information to prevent issues that arise when only one type is used.
Although the recommendation system has demonstrated its usefulness in various fields, it faces some problems: sparsity and cold start problems (CSP). As the name implies, a sparsity problem develops when customers do not rate an item while purchasing online, resulting in a sparsity of available ratings. The CF method goes through this problem because it uses a rating matrix. A CSP arises when a new product is added to the recommendation system, and no past ratings for that product are available. As a result, an RS must provide options for new users, which makes the accuracy of the recommendations low. Several research groups are working to develop an efficient and well-organised RS algorithm. The methods like local sensitive hashing (LSH) [13], Bayesian personalized ranking (BPR) [14], CofI Rank-maximum margin matrix factorization (CRMF) [15], etc. are used for removing sparsity and CSP. However, the existing methods fail to generate deep hashing by minimising classification loss, ranking pairwise, and high storage costs, motivating us to develop an efficient method for recommending items.

1.1. Contribution. The following are the research work’s contributions:

(i) The hash code is generated by reducing the classification and pairwise ranking loss, which can handle the users with cold start and sparsity problems.

(ii) A weight is introduced to hash tables and hash bits according to their performance.

(iii) The weighted Hamming distance is used to determine user similarity.

(iv) Bit diversity and similarity preservation are utilised to calculate the hash bit’s weight, and the MAP score is used to evaluate the weight-based table-wise.

(v) The proposed DRWMR system performs better for the item recommendation.

The following is a summary of the paper’s structure: Section 2 provides an overview of existing works. Our proposed DRWMR System is explained in Section 3. In Section 4, the proposed method is subjected to experimental analysis. Finally, the paper concludes in Section 5.

2. Literature Review

Da’u et al. [16] proposed a deep learning method that uses Aspect-based opinion mining (ABOM) in a recommendation system. This method contains two portions. They are rating predictions and ABOM. In the first portion, we utilised multichannel deep convolutional neural networks for extracting aspects. In the second portion, we add aspect-based ratings into a machine’s tensor factorisation to predict overall ratings. This method increases the accuracy of the recommendation system, but the time consumption is more significant.

Ye and Liu [17] introduced Collaborative Topic Regression (CTR) and three novel granulation methods for the recommendation strategy. The three granulation methods are LDA-based, PMF-based, and CTR-based, based on building granular structures and conducting granulation. LDA-based and PMF-based methods are developed to extract granular features from content and feedback information. The CTR-based method is designed to be joined with LDA- and PMF-based methods to produce multilevel recommendation information and interpretable granular features. These methods were introduced to overcome the problem of time cost and decision cost. However, this method does not consider users’ dynamic preferences.

Chen et al. [18] offered a Dynamic Decay CF (DDCF) recommendation method based on the user’s interest. This method has four stages: clustering of items, identification of the interesting level, specification of the decay function, and preference prediction. In item clustering, similar items are grouped without any predefined parameters. Then, in the second stage, based on the number of ratings and time records in the cluster, we can identify the interesting level of users. In the third stage, decay functions describe the preference evolution at every level. Finally, the similarities between the users are calculated based on the decay rates, and future preferences are predicted. This method accurately predicts the user’s interests, but the time consumption is high.

Beg et al. [19] introduced a chaotic based reversible data transform method (CRDTM) for preserving privacy in data mining for the recommendation system. This method will dynamically produce the values of RDT parameters during processing time, but it does not need the parameters during recovery. This method can substitute for the standard RDT algorithm, in which memory and bandwidth are considered significant factors. The RDT algorithm is used in the mobile app recommendation area. This method decreases the execution time, but accuracy is low.

Abbsi-Moud et al. [20] offered a context-aware recommendation system for tourism. It consists of three steps. ‘Users’ preferences are extracted from their reviews and comments in the first step. In the second step, tourist ‘attractions’ characteristics are extracted from the tourist reviews. In the third step, recommendations are given based on the ‘user’s preference and its similarity with the tourist attractions’ characteristics and contextual information. The contextual information utilised in this method includes location, weather, user preferences, and time. This method gives high accuracy in tourism recommendation. However, the weather may change according to the season, so it’s challenging to make a recommendation.

Zhang et al. [21] introduced a novel hybrid probabilistic matrix factorization method for distinguishing between items’ attractiveness and users’ preferences for a recommendation. It consists of two sub-divisions. One division attempts to predict the rating scores of users by extracting the user’s personal preferences from auxiliary data. Another division attempts to model the textual interest of items for different users. These two sub-divisions are formed into a unified framework using a global objective function. This method gives high accuracy.
Choe et al. [22] offered a hierarchical model based on a recurrent neural network (HMRNN) for considering users’ item usage histories in time series and sequences in a recommendation. This method contains two layers: a long-term (LT) and a short-term (ST). The ST sequences are handled by the ST layer, typically found in older ones. The LT layer collects the data from the preceding ST sequences and distributes it to the last time. This method increases the amount of stored user history. However, it does not take the consideration of temporal properties.

3. DRWMR System

We propose a novel deep ranking weighted multihashing recommender (DRWMR) system to suppress sparsity and CSP by considering CSP and sparsity issues. This method has two stages: the neighbours’ formation and recommendation phases. Initially, the user-item rating data of the user is fed into the neighbours’ formation stage, and deep convolutional neural networks are used to extract text features. The deep neural network consists of an additional layer (hash table). The hash codes are produced by reducing the classification loss and pairwise ranking loss. The similarity preservation is preserved by pairwise ranking loss, and the prediction error is minimised by classification loss. For a pair of users, the pairwise-loss function can be defined as follows:

\[
S(a_i, a_j, I_{i,j}) = \frac{1}{2}(1 - I_{i,j})H_d(a_i, a_j) + \frac{1}{2}I_{i,j}\max(1 - H_d(a_i, a_j), 0).
\]

Such that \(a_i, a_j \in \{+1, -1\}^K\),

where \(a\) denotes the user's binary hash code, \(H_d(a_i, a_j)\) signifies the Hamming distance between \(a_i\) and \(a_j\). The similarity between \(a_i\) and \(a_j\) is denoted by \(I_{i,j}\), it is defined as

\[
I_{i,j} = \begin{cases} 0 & \text{if } a_i \text{ and } a_j \text{ are similar} \\ 1 & \text{if } a_i \text{ and } a_j \text{ are dissimilar} \end{cases}
\]

The Hamming distance between the input user \(e_i\), and its pairwise user is calculated to produce an arranged hamming list. The pairwise ranking loss for an input user \(e_i\) is defined in the following equation:

\[
P(a_i) = \sum_{j=1}^{M_i} P(S(a_i, a_j, I_{i,j})) = \sum_{j=1}^{M_i} S(a_i, a_j, I_{i,j}),
\]

where \(M_i\) denotes the pairwise user’s number for user \(e_i\). Users similar to \(e_i\) should have short Hamming distances, therefore they seem at the top of the arranged Hamming list, whereas dissimilar data seem at the bottom. On the other hand, data at the improper positions of the Hamming list \(e_i\) has substantial loss values. We aim to decrease the overall loss function for a training database with \(M\) user, which is given in the following equation:

\[
P = \sum_{i=1}^{M} \sum_{j=1}^{M_i} P(a_i, a_j).
\]

Such that \(a_i, a_j \in \{+1, -1\}^K, i = 1, \ldots, M\).

Since the hash codes are in binary, the fitness function is nondifferentiable. As a result, using the gradient descent approach is challenging for optimising the fitness function.
Therefore, to overcome this problem, a tanh approximation function is used instead of the sign function. The tanh-like function can be used to estimate \( a_i \)'s hash code, which is given as follows:

\[
p_i(a_i) = \frac{e^{a_i} - e^{-a_i}}{e^{a_i} + e^{-a_i}}.
\]  

(6)

The Euclidean distance \( E_d(p_i, p_j) \) between two users, can be further estimated as the Hamming distance \( H(a_i, a_j) \) using the calculated hash codes determined in the following equation:

\[
E_d(p_i, p_j) = \|a_i - a_j\|_2.
\]  

(7)

A regularisation term is included to reduce the quantisation loss. The final loss function is determined in the following equation:

\[
P = \sum_{i=1}^{M} \sum_{j=1}^{M} \left\{ I_{i,j} E_d(p_i, p_j) + \frac{1}{2} I_{i,j} \max \left( 1 - E_d(p_i, p_j), 0 \right) \right\}^2 + \sum_{i=1}^{M} \|a_i - a\|_2.
\]  

(8)

Such that \( a_i, a_j \in \{+1, -1\}^h, i = 1, \ldots, \ldots, M \), where \( a \) denotes the regularisation term’s parameter coefficient. The classification layer outputs the recommendation for the category. As a recommendation function, the softmax function is used. The classification loss is determined in the following equation:

\[
C_i = -\sum_{j=1}^{o} D_{ij} e^{o_{xij}} \sum_{k=1}^{o} e^{o_{xik}}.
\]  

(9)

Where \( o \) and \( o_{xij} \) signify the class’s number and the recommended output of \( j \)'s class for user \( e_i \).

### 3.1.2. Weighted Multihashing

A weighted multihash code is used based on the loss in the hamming distance. It multiplies the bitwise weight and table-wise weight to minimise loss. Each hash bit performs differently in the recommendation task, so it is not fair to treat them all the same. Therefore, a weight is assigned to different hash tables and hash bits. Bit diversity and similarity preservation are integrated for the bitwise weighting. The MAP value is utilised for each hash table to compute the weight-based table-wise. The weight-based table-wise and bitwise may change for dissimilar users. This method adjusts the two weights, and their product determines the final weight. The bit \( b \) in table \( r \) final weight is defined as follows:

\[
WT_{br} = WT_{bit} \times WT_{bit}.
\]  

(10)

Similarity preservation (P) measures a hash bit’s semantic similarity. The hash bit \( a_i \)'s similarity preservation (P) is calculated in the following equation:

\[
Q(a_i) = \frac{1}{2} (a_i'a_i - U),
\]  

(11)

where \( U \) signifies the training sample’s number, and \( a_i' \) signifies the \( a_i \)'s transpose.

The bit diversity is utilised to calculate the hash bit’s performance. The difference between each user’s hash bit is crucial for maintaining recommendation efficiency. Therefore, every hash bit must be independent. The correlation between two hash bits is used to determine bit diversity, which can be determined by the above formula.

Initially, a bitwise correlation metric \( R = \{r_{Q_0} \mid i, j, \ldots, \ldots, h \} \in \mathbb{Z}^{h \times h} \) can be generated by calculating \( r_{Q_0} \) for bit \( a_i \) and \( a_j \) pair. The bitwise correlation is measured using the Kappa value, which is determined in the following equation:


\[ r_q = \frac{\emptyset_1 - \emptyset_2}{1 - \emptyset_2} \in [-1, 1], \]

where \( \emptyset_1 \) and \( \emptyset_2 \) denote the hash bit’s probabilities. Eqn (12) can determine the \( \emptyset_1 \) and \( \emptyset_2 \) probabilities.

\[
\emptyset_1 = \frac{w + z}{v}
\]

\[
\emptyset_2 = \frac{(w + x)(w + y) + (y + z)(x + z)}{v^2}
\]

where \( w, x, y, \) and \( z \) denote the nonnegative variables indicating the sample’s number that fulfills the relevant row and column requirements. The bit-correlated matrix \( M = \{m_1, \ldots, m_b\} \in \mathbb{Z}^{1 \times h} \) is formed by assigning the correlated coefficient for each bit. Finally, the bit\( a_i \)'s bit diversity weight is calculated by the following equation:

\[ K(a_b) = \frac{1 - f_b}{2}, \]

where \( f_b \) signifies the \( b^{th} \) bit’s correlated coefficient. To generate a final bitwise weight \( WT_{bi} \), the terms mentioned above are first adjusted and multiplied, which is determined in the following equation:

\[ WT_{bi} = Q(a_b) \times K(a_b). \]

To combine multiple hash tables, table-wise weight \( WT_{bi} \) for each hash table is determined using the mean average precision. It is specified in the following equation:

\[ A = \frac{1}{P} \sum_{b=1}^{n} \frac{1}{P_{b}} \times \text{rel}_{b}, \]

where \( n \) signifies the size of the dataset, \( P \) denotes the similar user’s number in the dataset that relates to the checking user, \( P_{b} \) denotes the similar user’s number at the top of the dataset.

3.2 Recommendation Phase. The neighbours obtained from the previous phase are used to generate a recommendation. This similarity between the active user and its neighbours is utilised to forecast the final rating for an unknown rating of item \( i \). In this regard, the ultimate anticipated rating provided by user \( u \) for any item \( i \) is calculated as the following:

\[ \text{Pre}_{u,i} = r_u + \frac{\sum_{u' \in N(u)} \text{sim}(u, u') \times (r_{u',i} - \bar{r}_u)}{\sum_{u' \in N(u)} \text{sim}(u, u')}, \]

where \( N(u) \) is the user’s neighbour, \( u \) and \( \text{sim}(u, u') \) denote the similarity between two users.

The prediction value of each item can be sorted in descending order, and the top \( n \) items from this sorted list of items can be recommended to the user.

The similarity between the users can be obtained by the following equation:

\[ \text{sim}(u, u') = \frac{\sum_{i=1}^{I} (r_{u,i} - \bar{r}_u)(r_{u',i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{I} (r_{u,i} - \bar{r}_u)^2 \sum_{i=1}^{I} (r_{u',i} - \bar{r}_u)^2}} \]

where \( I \) is the set of items, \( r_{u,i} \) rating of given to item \( i \) by user \( u \), \( \bar{r}_u \) average rating of user \( u \).

4. Experimental Results and Analysis

This section contains the experimental setup and a description of the datasets for comparative analysis.

4.1 Dataset Description. The movie Lens 100k dataset contains numerous users’ demographic data. There are 1682 movies in the dataset, with a total of 10,000 ratings from 943 users. For this evaluation, data were gathered from the https://www.kaggle.com/prajitdatta/movielens-100k-dataset. It consists of the user’s age, ID, occupation, and items provided. The datasets are used for 80% of the training, while the remaining 20% is used for testing.

4.2 Simulation Setup. The proposed DRWMR system is implemented in python; the initial learning rate is 0.001, and after 1000 iterations, it lowers exponentially by 0.04. The batch size is 200, and the parameter coefficient is 0.01.

4.3 Evaluation Criteria. The model’s performance is evaluated using six metrics: recall, precision, F1-score, MAE, RMSE, and prediction accuracy.

(i) Recall:

\[ \text{RL} = \frac{F_r}{T_r} \]

where \( F_r \) denotes the recommendation related to the new user and \( T_r \) signifies the real amount of related recommendations.

(ii) Precision:

This metric indicates the precision of the process, i.e., whether the generated recommendations are appropriate for new users.

\[ \text{PN} = \frac{F_r}{\text{tot}_r} \]

where \( F_r \) denotes the recommendation related to the new user and \( \text{tot}_r \) signifies the recommended item’s total quantity.
(iii) F1-score:
This score shows the experiment’s accuracy based on the recall and precision measures and is calculated by the following equation:
\[
F = \frac{2 \times RL \times PN}{PN + RL}
\]
(21)

(iv) RMSE:
It is measured by determining the difference between the predicted and observed values, which is given in the following equation:
\[
R = \sqrt{\frac{1}{Q} \sum_{j=1}^{Q} (k_j - \hat{k}_j)^2}
\]
(22)

(v) MAE:
The magnitude of the difference between the expected and observed values is computed to determine it, which is given in the following equation:
\[
M = \frac{1}{Q} \sum_{j=1}^{Q} |k_j - \hat{K}_j|
\]
(23)

where \( Q \) signifies the user’s total amount, \( \hat{K}_j \) denotes the predicted values, and \( K_j \) signifies the observed value.

4.4. Comparative Analysis with Existing Methods. The proposed DRWMR system is tested on the movie Lens 100K dataset, and metrics such as recall, precision, RMSE, MAE, and F1-score are compared to existing approaches such as CTR [17], DDCF [18], CRDTM [19], and HMRNN [22].

4.4.1. Quantitative Analysis. Figure 2 illustrates the RMSE analysis of our proposed DRWMR system with existing methods such as CTR, DDCF, CRDTM, and HMRNN. Our research introduces a novel loss function that minimises the classification loss and ranks pairwise losses. It produces hash codes with high recommendation accuracy and more similar information. Therefore, our proposed DRWMR system reduces the error compared to the existing methods for the top 5, 10, and 15 recommendations. The proposed DRWMR system (0.73) has a low RMSE when compared with existing techniques such as CTR (0.87), DDCF (0.80), CRDTM (0.77), and HMRNN (0.75) for the top 10 recommendations.

Figure 3 shows the MAE analysis. The smallest Hamming distance between the users results in the recommendation to the new user. The weighted Hamming distance is used to determine how similar the users are. So, the errors can be reduced in the recommendation system. The MAE is low for the proposed DRWMR system (0.57) when compared with existing methods such as CTR (0.75), DDCF (0.72), CRDTM (0.705), and HMRNN (0.653) for the top 10 recommendations.

Figures 4 and 5 show the recall and precision analyses. In the proposed DRWMR system, a hash table is built as an additional layer. A table and bitwise weight method are presented based on the performance to attain a better recommendation performance. The proposed DRWMR system assigns weights to various hash bits and hash tables. Also, a hash code is generated based on minimising loss,
which can handle users with cold start and sparsity problems. This mechanism attains higher precision and recall. The precision is high for the proposed DRWMR system (0.16) when compared with existing methods such as CTR (0.10), DDCF (0.12), CRDTM (0.129), and HMRNN (0.14) for the top 10 recommendations. The recall is high for the proposed DRWMR system (0.08) when compared with existing methods such as CTR (0.04), DDCF (0.041), CRDTM (0.045), and HMRNN (0.06) for the top 10 recommendations.
Figure 5: Recall analysis.

Figure 6: F-measure analysis.
Figure 6 shows the $F$–1 measure analysis. The $F$-measure is high for the proposed DRWMR system (0.101), when compared with existing methods such as CTR (0.06), DDCF (0.064), CRDTM (0.077), and HMRNN (0.08) for the top 10 recommendations.

5. Conclusion

The RS has become vital to social networking and business apps such as Flipkart, Amazon, YouTube, and others. Therefore, we introduced the DRWMR system for an accurate recommendation initially. User-item rating data is fed into CNN to extract important features to reduce data sparsity. Then, the hash code is generated by minimising pairwise rank loss and classification loss. A weight is assigned to different hash tables and hash bits. According to weighted Hamming distance, the most similar users were obtained. Finally, the rating of unknown items can be obtained using the weighted average rating of similar users and active users. As a result, we have seen that our proposed DRWMR system performs better in recommending items. The proposed DRWMR system is quantitatively measured by precision (0.16), recall (0.08), RMSE (0.73), MAE (0.57), and F-measure (0.101). It shows better performance in the recommendation system.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, “Collaborative filtering and deep learning based recommendation system for cold start items,” Expert Systems with Applications, vol. 69, pp. 29–39, 2017.
[2] M. Kawasaki and T. Hasuike, “A recommendation system by collaborative filtering including information and characteristics on users and items,” in Proceedings of the In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 2017 November.
[3] D. Das, L. Sahoo, and S. Datta, “A survey on recommendation system,” International Journal of Computer Application, vol. 160, no. 7, pp. 6–10, 2017.
[4] P. Lopes and B. Roy, “Dynamic recommendation system using web usage mining for e-commerce users,” Procedia Computer Science, vol. 45, pp. 60–69, 2015.
[5] S. Pandya, J. Shah, N. Joshi, H. Ghayvat, S. C. Mukhopadhyay, and M. H. Yap, “A novel hybrid based recommendation system based on clustering and association mining,” in Proceedings of the In2016 10th International Conference on Sensing Technology (ICST), IEEE, Nanjing, China, 2016 November.
[6] J. P. Verma, B. Patel, and A. Patel, “Big data analysis: recommendation system with Hadoop framework,” in Proceedings of the 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, UP, India, 2015 February.
[7] S. Shaikh, S. Rath, and P. Janrao, “Recommendation system in E-commerce websites: a graph-based approached,” in Proceedings of the In 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, India, 2017 January.
[8] T. Rutkowski, J. Romanowski, P. Woldan, P. Staszewski, R. Nielek, and L. Rutkowski, “A content-based recommendation system using neuro-fuzzy approach,” in Proceedings of the In2018 IEEE International Conference on Fuzzy Systems, pp. 1–8, Hyderabad, India, 2018 July.
[9] S. Parvatikar and B. Joshi, “Online book recommendation system by using collaborative filtering and association mining,” in Proceedings of the In 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICICCI), pp. 1–4, Trivandrum Kerala, 2015 December.
[10] W. T. Chu and Y. L. Tsai, “A hybrid recommendation system considering visual information for predicting favorite restaurants,” World Wide Web, vol. 20, no. 6, pp. 1313–1331, 2017.
[11] G. Suganeshwari and S. S. Ibrahim, “A survey on collaborative filtering based recommendation system,” in Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16’), Springer Cham, India, 2016 February.
[12] J. Jooa, S. Bangh, and G. Parka, “Implementation of a recommendation system using association rules and collaborative filtering,” Procedia Computer Science, vol. 91, pp. 944–952, 2016.
[13] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, “Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing,” IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1145–1153, 2021.
[14] G. Li, T. Zhu, J. Hua et al., “Asking images: hybrid recommendation system for tourist spots by hierarchical sampling statistics and multimodal visual bayesian personalized ranking,” IEEE Access, vol. 7, Article ID 1265, 2019.
[15] M. Weimer, A. Karatzoglou, Q. Le, and A. Smola, “Cofirmatrix margin matrix factorisation for collaborative ranking Advances in Neural Information Processing Systems,” in Proceedings of the 21st Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2007 December.
[16] A. Da’u, N. Salim, I. Rabiu, and A. Osman, “Recommendation system exploiting aspect-based opinion mining with deep learning method,” Information Sciences, vol. 512, pp. 1279–1292, 2020.
[17] X. Ye and D. Liu, “An interpretable sequential three-way recommendation based on collaborative topic regression,” Expert Systems with Applications, vol. 168, Article ID 114454, 2021.
[18] Y. C. Chen, L. Hui, and T. Thapisutikul, “A collaborative filtering recommendation system with dynamic time decay,” The Journal of Supercomputing, vol. 77, no. 1, pp. 244–262, 2021.
[19] S. Beg, A. Anjum, M. Ahmed et al., “Dynamic parameters based reversible data transform (RDT) algorithm in recommendation system,” IEEE Access, vol. 9, pp. 110011–110025, 2021.
[20] Z. Abbasi-Moud, H. Vahdat-Nejad, and J. Sadri, “Tourism recommendation system based on semantic clustering and sentiment analysis,” *Expert Systems with Applications*, vol. 167, Article ID 114324, 2021.

[21] X. Zhang, H. Liu, X. Chen, J. Zhong, and D. Wang, “A novel hybrid deep recommendation system to differentiate user’s preference and item’s attractiveness,” *Information Sciences*, vol. 519, pp. 306–316, 2020.

[22] B. Choe, T. Kang, and K. Jung, “Recommendation system with hierarchical recurrent neural network for long-term time series,” *IEEE Access*, vol. 9, pp. 72033–72039, 2021.