

Retraction

Retracted: Personalized Book Recommendation Algorithm for University Library Based on Deep Learning Models

Journal of Sensors

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

Copyright © 2023 Journal of Sensors. 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This article has been retracted by Hindawi following an investigation undertaken by the publisher [1].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) Manipulated or compromised peer review

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] D. Hou, "Personalized Book Recommendation Algorithm for University Library Based on Deep Learning Models," *Journal of Sensors*, vol. 2022, Article ID 3087623, 6 pages, 2022.

Research Article

Personalized Book Recommendation Algorithm for University Library Based on Deep Learning Models

Dongjin Hou 

Library, People's Public Security University of China, Beijing 100038, China

Correspondence should be addressed to Dongjin Hou; houdongjin@ppsuc.edu.cn

Received 2 June 2022; Accepted 14 June 2022; Published 22 June 2022

Academic Editor: Yaxiang Fan

Copyright © 2022 Dongjin Hou. 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.

Personalized recommendation is one of the important contents of personalized service in university libraries. Accurate and in-depth understanding of users is the premise of personalized recommendation. This paper proposes a personalized book recommendation algorithm based on deep learning models according to the characteristics and laws of user savings in university libraries. The method first uses the long short-term memory network (LSTM) to improve the deep autoencoder (DAE) so that the model can extract the temporal features of the data. Then, the Softmax function is used to obtain the book recommendation result of the current user. The proposed method is verified based on actual library lending data. The experimental results show that the proposed method has performance advantages compared with several existing recommendation methods.

1. Introduction

The application of “Internet +,” social network, and other technologies in the library provides multisource data for user analysis, intelligent, and personalized digital library resource recommendation services, which will greatly improve users’ learning efficiency and experience. It is a meaningful work to help the scholars focus on how to mine users’ preferences and interests so as to satisfy them through recommendation services. Therefore, personalized digital library resource recommendation technology has gradually become a hot research point in the field of education and intelligence [1–4].

Early recommendation systems mainly used collaborative filtering (CF) algorithms. The main types of CF algorithms can be divided into user-based and item-based ones. Both types of algorithms need to be based on the constructed binary co-occurrence matrix of users and items and cooperate with the entire matrix data to predict the user’s response to the item. User-based CF needs to calculate the similarity between users and find users similar to the target user. Then, the weighting process is performed to sum up the scores of similar users as the target user’s predicted score for the item. The scores are sorted to

generate a list of recommended items. Reference [5] proposed a recommendation method based on CF, which mainly recommended products based on preferences with other users who like the item. Based on the recommendation accuracy evaluation indicators, CF recommendation has been greatly improved and gradually accepted by academia and business circles. However, this recommendation method only utilizes shallow model dependencies, which cannot learn deep feature representations for users and items, and lacks interpretability compared with many content-based algorithms. Based on the above problems, many researchers have improved the CF recommendation algorithm from different aspects. Literature [6] further combined the latent factor model (LFM) with CF, which made the CF algorithm a step forward. Literature [7] proposed a CF method based on matrix factorization (MF) and its variant algorithm, and experimental results showed that this method was more effective in rating prediction. However, the “latent factors” in latent models were not intuitively explanatory, making it difficult to understand why an item predicts a higher rating and is recommended. Although the traditional content-based recommendation methods can improve the interpretability of the recommendation, it extracts user-item features

based on a shallow model for recommendation. This feature extraction method relies heavily on artificially designed features, which seriously restricts the effectiveness and scalability of recommendation.

With the development of computer technology, artificial intelligence technology, and other fields, deep learning technology has been widely used in speech recognition, image processing, etc. This technology can not only accurately obtain the relationship between users and items in diverse data, but also can convert abstract codes into high-level data information. Therefore, deep learning technology is widely used in recommender systems. In [8], the deep learning model was fused with CF and applied in movie recommendation. Inspired by the good performance, more and more researchers have carried out research on deep learning network models applied to recommender systems. Literature [9] proposed a recommendation model based on long short-term memory network (LSTM) for course association classification. First, the generalized sequential pattern mining algorithm and spectral clustering algorithm were used to group courses, and then LSTM was used to model the time series of courses. Finally, the courses are classified using the Softmax function. The experimental results showed that the proposed algorithm has higher recommendation accuracy than other algorithms such as CF. Reference [10] proposed a recommendation model based on gated recurrent unit (GRU) and the relationship between courses. Combined with the relationship between courses, GRU and Softmax function were used to recommend courses. The experimental results showed that compared with the recommendation algorithm using CF, the recommendation accuracy rate was much higher. Reference [11] employed an autoencoder to map data into a nonlinear latent space, and experimental results showed that it had a lower mean square error than CF. Reference [12] combined the evaluation information and auxiliary information such as user attributes and item descriptions and used autoencoders to decompose the matrix. It also verified the effectiveness of the method through experiments.

Due to the powerful nonlinear mapping ability of deep learning when processing high-dimensional data, it can be effectively mapped to low-dimensional space to extract high-level features. The recommendation has also been widely used [13–16]. Due to the powerful latent feature learning ability of deep autoencoder (DAE), it can effectively deal with the problem of data sparsity [17]. In addition, because users often consider the previous book when borrowing new books, the user's borrowing sequence has obvious time series characteristics. LSTM has a strong time series modeling ability, which can effectively process time series data. According to the above characteristics, this paper uses LSTM to improve the DAE and applies it to personalized book recommendation in university libraries. The proposed method recommends future borrowed books according to the user's borrowing history [18–20]. The experiment is carried out based on the real loan data of the library, and the proposed method is compared with several existing prediction models. The experimental results show

that the recommended results of the proposed method are closer to the actual results, reflecting its higher effectiveness.

2. Method Description

2.1. LSTM. The borrowed data of readers has a certain time sequence. According to the characteristics of the neural network, this paper chooses the LSTM as the basic model to perform deep learning on the borrowed time series data set, which can maximize the data between time series and nonlinearity relationship [17–20]. LSTM makes the information selectively affect the state of each moment in the model by adding a gate structure, which is mainly composed of input gate, output gate, and forgetting gate. The basic structure of the LSTM unit is shown in Figure 1.

Assuming that the input sequence is $[x_1, x_2, \dots, x_t]$, and the hidden layer state is $[h_1, h_2, \dots, h_t]$, then there is the following mathematical expression at time t .

2.1.1. Forgetting Gate Operation. In LSTM, a certain probability is used to control whether to forget the hidden cell state passed to the previous layer, and the sigmoid function is used to control the output range between $[0, 1]$. Equation (1) displays the details:

$$f_t^l = \sigma \left(W_f^l \cdot [h_{t-1}^l, x_t^{l-1}] + b_f^l \right), \quad (1)$$

where σ is the activation function; \cdot represents the vector inner product; t represents the time; l represents the number of layers of the LSTM; f represents the forget gate; w represents the weight; b represents the bias; and h represents the cell output.

2.1.2. Input Gate Operation. The input gate consists of two parts. The sigmoid and tanh activation functions are used to control the range of the output value, and the product of the two parts is used to participate in the update of the cell state. Equations (2) and (3) show the details:

$$g_t^l = \sigma \left(W_g^l \cdot [h_{t-1}^l, x_t^{l-1}] + b_g^l \right), \quad (2)$$

$$\tilde{c}_t^l = \tanh \left(W_c^l \cdot [h_{t-1}^l, x_t^{l-1}] + b_c^l \right). \quad (3)$$

In the above equations, $\tanh()$ represents the activation function; g represents the input gate; c represents the cell state; and \tilde{c} represents the current input cell state.

2.1.3. Cell State Update. The cell state is updated by calculating the product of the cell state at the previous moment, the output of the forget gate, and the product of the two parts of the input gate. The products of the two parts are added to update the cell state:

$$c_t^l = f_t^l * h_{t-1}^l + g_t^l * \tilde{c}_t^l, \quad (4)$$

where $*$ represents the multiplication of elements of the corresponding dimension of the two vectors.

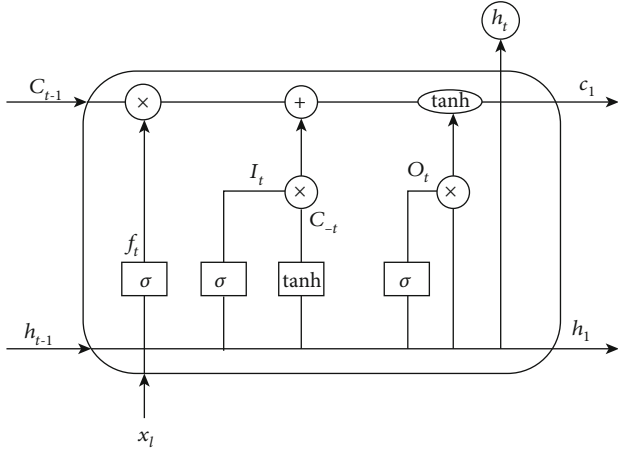


FIGURE 1: Basic structure of LSTM.

2.1.4. Output Gate Operation. The output gate consists of two parts. The first part also uses the hidden state at the previous moment and the input variable at this moment as input, and the output range is controlled by the sigmoid function. The second part uses the hidden state to control the output through the tanh activation function, which is multiplied with the output result of the first part to update the hidden state:

$$o_t^l = \sigma \left(W_o^l \cdot [h_{t-1}^l, x_t^{l-1}] + b_o^l \right), \quad (5)$$

$$h_t^l = h_t^l \cdot \tanh c_t^l, \quad (6)$$

where o represents the output gate.

2.2. DAE. DAE is composed of a stack of multiple sparse autoencoders. The sparse autoencoder draws on the neuron excitability mechanism of the brain. After encoding, the original data can be decoded to the maximum extent, and it also has the advantages of fast convergence speed and avoiding falling into local minima and other advantages [14–16]. The basic structure of DAE is shown in Figure 2.

The stacked autoencoder adopts a layer-by-layer greedy learning method, taking the hidden layer of the trained first sparse autoencoder as the input of the second sparse autoencoder. The hidden layer of the second sparse autoencoder is also used as the input. The input of the next sparse autoencoder is to train the sparse autoencoder in turn, and the hidden layer data of the last sparse autoencoder is used as the output of the entire stack autoencoder network. In general, the hidden layer data of each sparse autoencoder can be regarded as a potential feature of the original training data. Through this feature, the feature information contained in the original data can be decoded to the greatest extent. Therefore, multiple sparse autoencoders can be regarded as a process of extracting high-dimensional latent features of the original data.

The sparse autoencoder controls the complexity of the model by optimizing the weights and the average activation of neurons in the hidden layer to reduce network overfitting.

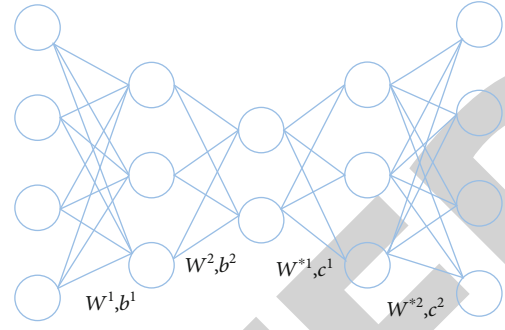


FIGURE 2: Basic structure of DAE.

It can be realized by adding L2 regularization term and sparse term to its cost function:

$$C = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K (x_{kn} - \hat{x}_{kn})^2 + \lambda \cdot \Omega_{\text{weight}} + \beta \cdot \Omega_{\text{sparsity}}, \quad (7)$$

where x_{kn} is the input sample data; \hat{x}_{kn} is the actual output value; Ω_{weight} and Ω_{sparsity} are the L2 regularization term and the sparse term, respectively; λ and β are the coefficients of the L2 regularization term and the sparse term, respectively; N is the total number of samples; and K is the length of the input signal.

The essence of L2 regularization is the penalty on the weight vector, which can suppress the larger value in the weight and make the network tend to learn the smaller weight:

$$\Omega_{\text{weight}} = \frac{1}{2} \sum_l \sum_j^n \sum_i^k \left(w_{ji}^{(l)} \right)^2. \quad (8)$$

The function of the sparse term is to control the activation number of hidden neurons. Generally, if the output of a neuron is close to 1, the neuron is considered to be activated. Otherwise, if the output is 0, it is considered to be inhibited. The purpose of sparse autoencoders is to keep these neurons inactive most of the time. Assuming that $z_i(x_j)$ is the i th activation unit of the hidden layer during the forward propagation process, the activation unit of the hidden layer can be expressed as $h(w_i^T + b_i)$, in which w is the weight matrix connecting the input layer and the hidden layer, and b is the offset between the two layers. Then, the hidden layer average activation of the i th unit is as follows:

$$\hat{\rho}_i = \frac{1}{n} \sum_{j=1}^n z_i(x_j) = \frac{1}{n} \sum_{j=1}^n h(w_i^T + b_i). \quad (9)$$

Since sparsity requires most neurons to be suppressed, it is hoped that the average activation $\hat{\rho}_i$ is close to a constant ρ approximately 0, which is the sparsity parameter. In order to achieve such a sparsity effect, an additional penalty term is added to the cost function to make the average activation $\hat{\rho}_i$ not deviate from the sparsity parameter ρ . Accordingly, the

Kullback-Leibler (KL) divergence is chosen to achieve the purpose of punishment. The sparse regularization term can be expressed as follows:

$$\Omega_{\text{sparsity}} = \sum_{i=1}^D KL(\rho \parallel \hat{\rho}_i) = \sum_{i=1}^D \rho \log \left(\frac{\rho}{\hat{\rho}_i} \right) + (1 - \rho) \log \left(\frac{1 - \rho}{1 - \hat{\rho}_i} \right). \quad (10)$$

In Equation (10), D is the number of hidden layer units. If $\hat{\rho}_i = \rho$, then $KL(\rho \parallel \hat{\rho}_i) = 0$.

To sum up, the cost function of sparse autoencoder can be expressed as follows:

$$C = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K (x_{kn} - \hat{x}_{kn})^2 + \frac{\lambda}{2} \sum_l \sum_j \sum_i (w_{ji}^{(l)})^2 + \beta \sum_{i=1}^D KL(\rho \parallel \hat{\rho}_i). \quad (11)$$

Finally, the entire network obtains the optimized weights w and bias b through the back-propagation algorithm.

3.2. Book Recommendation Model. In order to model the temporal sequence of the user's book borrowing data, LSTM is used to replace the feedforward neural network in DAE, and the two are combined. The basic process of the proposed method is shown in Figure 3. The time series feature extraction method proposed in this paper includes five parts: input layer, encoding layer, LSTM layer, decoding layer, and output layer. First, the input data is preprocessed, including data enhancement and sequence division by window. And then each sequence is sent to the encoder to obtain a feature vector, which is then used as the input of the LSTM network. Finally, the output of the LSTM is sent to the decoder to obtain a reconstruction of the next window of the original data window. The training process is performed by minimizing the reconstruction error. And after the training is completed, the LSTM network output is extracted as the extracted time series features. After the time series features of the data are extracted in the previous step, the output of the LSTM is used as the input, and the final recommendation result of the model is obtained through the Softmax function.

3. Experiment and Analysis

3.1. Sample Set and Preprocessing. A real data set of a library is used, which contains 342,451 book loan records, including 4,652 users and 6,783 books. Each record contains attributes such as user id, borrowing time, book id, book name, and book category.

Step 1: The original data is simplified into triples (user id, course selection time, and course id)

Step 2: From the 4652 users, the users who have borrowed less than 5 books are removed, leaving 184201 pieces of data and 3347 users

Step 3: According to the user grouping, it is divided into 2675 groups. In each group, the books are sorted in ascending order according to the time of borrowing books

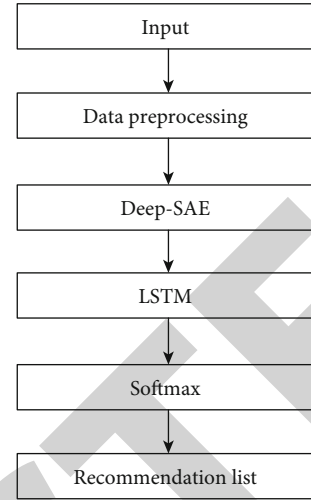


FIGURE 3: Basic procedure of book recommendation model.

Step 4: The data in the group is merged according to (user id, book id time series) to get the final data set, which contains a total of 3789 pieces of data

3.2. Evaluation Index. In this paper, three evaluation indexes commonly used in Top- N recommender systems, i.e., precision (P), recall (R), and F_1 score (F_1), are used to measure the recommendation performance of the proposed method, which are calculated as follows:

$$P = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|},$$

$$R = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}, \quad (12)$$

$$F_1 = \frac{2 \times P \times R}{P + R}.$$

where $R(u)$ represents the book recommendation list made to the user based on the book borrowing data on the training data set and $T(u)$ represents the user's book borrowing list on the test data set.

3.3. Result and Analysis. For the Top- N recommendation problem, this paper takes 3 groups of comparative experiments at $N \in \{1, 5, 10\}$. In each group of comparative experiments, the CF, autoencoder (AE), and LSTM methods in the existing methods are selected for comparison. Among them, the LSTM method only uses LSTM for recommendation, and no other deep learning models are introduced.

Tables 1, 2, and 3 show the performance comparison of various methods under different recommendation numbers. Under the same conditions, the proposed method outperforms the three comparison algorithms in terms of P , R , and F_1 . In particular, compared with LSTM, this paper further improves the recommendation performance by incorporating DAE, demonstrating the effectiveness of the model. The experimental results show that the performance of the recommendation algorithm based on the deep

TABLE 1: Comparison of different methods at $N = 1$.

Method	P	R	F_1
Proposed	0.172	0.158	0.164
CF	0.147	0.141	0.145
AE	0.156	0.152	0.154
LSTM	0.163	0.158	0.160

TABLE 2: Comparison of different methods at $N = 5$.

Method	P	R	F_1
Proposed	0.398	0.382	0.387
CF	0.356	0.348	0.361
AE	0.369	0.354	0.363
LSTM	0.378	0.365	0.371

TABLE 3: Comparison of different methods at $N = 10$.

Method	P	R	F_1
Proposed	0.543	0.528	0.532
CF	0.497	0.482	0.488
AE	0.518	0.503	0.508
LSTM	0.524	0.516	0.519

learning is better than that of the traditional algorithms, because the deep learning algorithm can extract deeper data features. The method proposed in this paper uses DAE for deep feature extraction and applies LSTM to model the time series features of the data, so the final overall performance is further improved.

Comparing the results under different recommended numbers, it can be seen that with the increase of N , the three performance evaluation indexes of various methods are also getting higher and higher. The probability is also increasing. Compared with various methods, the improvement of the method in this paper is more significant, which further reflects its effectiveness.

4. Conclusion

This paper proposes a recommendation model based on DAE and LSTM for the personalized book recommendation problem in university libraries. First, LSTM is used to improve the DAE so that the model can extract the temporal characteristics of the data. The Softmax is used for the direct recommendation of books for the users. Finally, the experimental verification is carried out on the real library lending data set. The experimental results show that the proposed method has higher accuracy compared with several existing recommendation algorithms. Therefore, the method in this paper can be applied in subsequent library lending systems to enhance overall work efficiency.

Data Availability

The data set can be accessed upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

- [1] L. Zuo, S. Xiong, X. Qi, W. Zheng, and Y. Tang, "Communication-based book recommendation in computational social systems," *Complexity*, vol. 2021, Article ID 6651493, 10 pages, 2021.
- [2] W. Yang, "Personalized intelligent recommendation algorithm design for book services based on deep learning," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 9203665, 8 pages, 2022.
- [3] Z. H. Wang and D. Z. Hou, "Research on book recommendation algorithm based on collaborative filtering and interest degree," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7036357, 7 pages, 2021.
- [4] Y. Shi and Y. Zhu, "Research on fast recommendation algorithm of library personalized information based on density clustering," *Security and Communication Networks*, vol. 2022, Article ID 1169115, 9 pages, 2022.
- [5] S. Zhang, L. N. Yao, A. X. Sun, and Y. Tay, "Deep learning based recommender system: a survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, p. 5, 2020.
- [6] B. Yang, Y. Lei, J. M. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.
- [7] L. H. Jonathan, A. K. Joseph, and R. John, "Explaining collaborative filtering recommendations," in *ACM Conference on Computer Supported Cooperative Work*, pp. 241–250, Hangzhou, China, 2011.
- [8] F. Strub, R. Gaudel, and J. Mary, "Hybrid recommender system based on autoencoders," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 11–16, Boston, 2016.
- [9] J. Li and Z. Ye, "Course recommendations in online education based on collaborative filtering recommendation algorithm," *Complexity*, vol. 2020, Article ID 6619249, 10 pages, 2020.
- [10] M. Qin, H. Chen, R. Zheng, and T. Gao, "An adaptive operational modal analysis method using encoder LSTM with random decrement technique," *Journal of Sensors*, vol. 2021, Article ID 7199888, 11 pages, 2021.
- [11] X. Liu, "Music trend prediction based on improved LSTM and random forest algorithm," *Journal of Sensors*, vol. 2022, Article ID 6450469, 10 pages, 2022.
- [12] Z. H. Cui, X. H. Xu, F. Xue et al., "Personalized recommendation system based on collaborative filtering for IoT scenarios," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 685–695, 2020.
- [13] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: autoencoders meet collaborative filtering," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 111–112, Florence, 2015.
- [14] J. Zhang, B. W. Hao, B. Chen, C. Li, H. Chen, and J. Sun, "Hierarchical reinforcement learning for course recommendation in MOOCs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 435–442, 2019.
- [15] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, no. 27, pp. 297–305, 2018.

- [16] H. Fang, G. Guo, D. Zhang, and Y. Shu, "Deep learning-based sequential recommender systems: concepts, algorithms, and evaluations," in *International Conference on Web Engineering*, pp. 574–577, Springer, Cham, 2019.
- [17] Q. Tan and F. Liu, "Recommendation based on users' long-term and short-term interests with attention [J]," *Mathematical Problems in Engineering*, vol. 2019, 13 pages, 2019.
- [18] S. Xing, F. Liu, X. Zhao, and T. Li, "Content-aware point-of-interest recommendation based on convolutional neural network [J]," *Applied Intelligence*, vol. 48, no. 8, pp. 2458–2469, 2018.
- [19] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trustaware recommendations in social networks [J]," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1164–1177, 2017.
- [20] X. Hu, R. Men, G. Li, and Z. Jin, "Deep-AutoCoder: learning to complete code precisely with induced code tokens," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 159–168, Milwaukee, WI, USA, 2019.