# Research on Movie Recommendation Model Based on LSTM and CNN

Wentao Wang
School of Computing
Qinghai Normal University
Xining 810000, China
e-mail: 931826985@qq.com

Chengxu Ye *
School of Computing
Qinghai Normal University
Xining 810000, China
e-mail: 149926237@qq.com

Ping Yang
School of Geographical Sciences
Qinghai Normal University
Xining 810000, China
e-mail: 417666733@qq.com

Zhikun Miao
School of Computing
Qinghai Normal University
Xining 810000, China
e-mail: 841058010@qq.com

***Abstract*—In order to further improve the accuracy of movie recommendation, while considering the characteristics of user data and movie data, this paper studies and proposes a combined recommendation model of LSTM and CNN. The model uses LSTM to capture the context dependency of user ratings data, and at the same time extracts the local relevant features of the movie title with CNN, and then fuse each feature to calculate the predicted ratings, through model training and optimization, the movie recommendation to the user is finally obtained according to the predicted ratings. The MovieLens data set is used to verify the effectiveness of the model, and the results show that compared with the traditional recommendation model and other recommendation models based on deep learning, the combined recommendation model of LSTM and CNN proposed in this paper have a MSE loss reduction of 4.4%~18.7% and a MAE loss reduction of 3.0%~52.2%.**

***Keywords- rating prediction; movie recommendation; LSTM; CNN***

## I. INTRODUCTION

With the rapid development of Internet technology, the scale of data is increasing exponentially, and the phenomenon of "information overload" is profoundly affecting every Internet user [1]. As a method to effectively solve the above problems, the recommendation model has been widely used in actual production environments, such as information retrieval, e-commerce, news media and other fields [2].

In the field of movie and television, the recommendation model has also become the most powerful driver for the development of major video platforms. For example, the video site YouTube claims that of the total time users spend watching videos, the time spent watching recommended content is as high as 70%. The video website Netflix also organizes related competitions to find more accurate models for video recommendation, and continuously improves its own recommendation model [2]. Therefore, designing an efficient and accurate movie recommendation model will not only solve the "information overload" problem in the field of movie and television, but also promote the development of the movie and television industry.

## II. RELATED RESEARCH

### *A. Research Status of Recommendation Model*

Traditional recommendation models mainly include collaborative filtering recommendation models, content-based recommendation models and hybrid recommendation models [2]. Collaborative filtering is currently the most widely recommendation model, but it faces serious data sparseness and cold start problems. The content-based recommendation model requires effective feature extraction, but the traditional shallow model relies on artificial design features, which restricts the performance of the content-based recommendation model [2]. At the same time, the existing traditional recommendation model cannot reasonably model the serialized user behavior data, and it is difficult to learn the context dependency in the sequence data.

Many studies have introduced deep learning into the construction of recommendation models, and achieved good recommendation results. For example, Yuyun Gong and Qi Zhang proposed a convolutional neural network (CNN) label recommendation model based on the attention mechanism in 2016. Compared with traditional recommendation models, this model performs well [3]. Jeffrey Lund and Yiu-Kai Ng proposed a deep learning model based on Autoencoder (AE) in 2018 to predict users' ratings of new movies to achieve movie recommendation. The model was also obtained in the experiment Better results [4].

The deep learning-based recommendation model can represent the relationship between users and items by learning a deep-level nonlinear network structure, has a strong ability to learn the essential characteristics of data from samples, and can obtain deep-level feature representations of users and items [2]. However, existing recommendation models based on deep learning do not take into account the effective extraction of user features and item features. Therefore, this paper proposes a combined recommendation model of Long-short Term Memory

978-1-7281-6042-9/20/$31.00 ©2020 IEEE
DOI 10.1109/ICCIA49625.2020.00013

Network (LSTM) and CNN. The model will effectively extract user features and item features, calculate prediction ratings and generate recommendation lists.

### B. LSTM Related Research

LSTM is improved from Recurrent Neural Network (RNN). LSTM introduced the concept of gates while adding memory cells to the structure [5]. Due to the unique structural design, LSTM is often used to analyze and process time series data with large intervals and delays. Figure 1 shows the structure of the LSTM model.

LSTM implements three gate calculations: forget gate, input gate and output gate. LSTM neurons read, output and update long-distance historical information by these three gates [5].

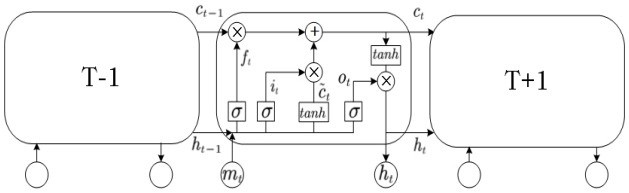


Figure 1. LSTM model structure figure.

Let $m_t$ be the input of the current time, $c_t$ be the state of the hidden layer at the current time, and $h_t$ be the output of the cell state at the current time. The forget gate is responsible for controlling the amount of information that the cell state $c_{t-1}$ at the previous moment retains to the cell state $c_t$ at the current moment. The input gate is responsible for controlling how much of the current input $m_t$ to the current cell state $c_t$ is reserved. The output gate obtains the cell state output $h_t$ at the current time based on the new cell state $c_t$.

LSTM is usually calculated as follows:

$$f_t = \sigma(W_f[m_t, h_{t-1}] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[m_t, h_{t-1}] + b_i) \quad (2)$$

$$\tilde{c}_t = \tanh(W_c[m_t, h_{t-1}] + b_c) \quad (3)$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet \tilde{c}_t \quad (4)$$

$$o_t = \sigma(W_o[m_t, h_{t-1}] + b_o) \quad (5)$$

$$h_t = o_t \bullet \tanh(c_t) \quad (6)$$

Among them: σ and tanh are respectively sigmoid activation function and hyperbolic tangent activation function; W is the weight coefficient matrix; b is the corresponding offset term; $\tilde{c}_t$ is the accumulated information at the current moment.

### C. CNN Related Research

CNN is a typical feedforward neural network, and its special design greatly reduces the complexity of the neural network structure, making it achieve good results in many fields [6].

As shown in Figure 2, the structure of CNN includes: input layer, convolution layer, pooling layer, fully connected layer and output layer. The convolution operation and the pooling operation are alternated, a pooling layer immediately follows a convolutional layer, and usually these two layers will take several according to actual needs [7].

In the convolutional layer, let current input be $X_i$ , and $C_i$ is the current calculation result of the convolution operation. The weight coefficient matrix of the layer i convolution kernel is represented by $W_i$, using the symbol "⊙" to represent the convolution operation, which is performed between the current inputs $X_i$ and $W_i$. After the convolution operation, the output value is added to the offset vector $b_i$ of the ith layer, and then the excitation function f(x) is used to obtain $C_i$. The calculation procedure of $C_i$ is described as equation (7).

$$C_i = f(X_i \odot W_i + b_i) \quad (7)$$

The pooling layer adopts the corresponding rules to sample the feature map to reduce and abstract the features. There are many common pooling rules, including maximum pooling, average pooling, and random pooling [7]. Pooling operations are described as equation (8).

$$P_i = pooling(C_i) \quad (8)$$

After going through several convolution and pooling operations, the result will be passed to the fully connected layer. Each neuron in the fully connected layer is fully connected with all the neurons in the previous layer. Through this connection, the part of the information with high class distinction in the features is fully integrated [8]. To further optimize the performance of the CNN network, the excitation function of the fully connected layer generally adopts the ReLU function, which is defined as equation (9). Finally, the output value is obtained by using softmax logistic regression calculation in the output layer.

$$f(x) = \max(0, x) \quad (9)$$

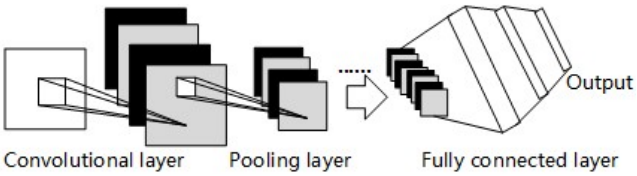


Figure 2. CNN model structure figure.

## III. The Combined Recommendation Model of LSTM and CNN

LSTM and CNN use special structural design to deal with different problems. CNN uses the convolution kernel as an intermediary to fully mine the local information of the input data. However, due to the size constraints of the convolution kernel, CNN cannot effectively extract the context of events in time series data [9]. LSTM uses gate calculation and memory cells to effectively analyze and process time series data with large intervals and delays [10]. Therefore, this paper selects LSTM to process the time-series rating data to extract user preference features. For the movie title, the model selects CNN to capture the locally relevant features. The combined recommendation model of LSTM and CNN proposed in this paper will further combine the advantages of the two deep learning methods and make the two methods complement each other in the shortcomings.

The combined recommendation model of LSTM and CNN designed in this paper is shown in Figure 3. The model takes user information, rating data and movie data as input to

learn user features and movie features. Next, the model calculates prediction ratings and makes movie recommendations based on these two features. The model mainly includes three parts: input layer, model layer and output layer. The input layer data mainly includes: user information (such as user ID, gender, age, occupation, etc.), user rating data (such as user ID, movie ID, rating, timestamp, etc.) and movie data (such as movie ID, movie title, movie genres, etc.). At the model layer, for the user information and movie data, the embedding method is used to obtain the features of the corresponding fields, use CNN to mine the local information of the movie title, and then the fully connected network is used to fully integrate the features to obtain the user explicit feature $U_{ef}$ and the movie feature $M_f$. For user rating data, first filter and sort, then use LSTM to combine the obtained movie features to model the sequence pattern of user rating data, obtain user preference feature $U_{pf}$. Then combine user explicit feature $U_{ef}$ and user preference feature $U_{pf}$ get user features $U_f$. At the output layer, a fully connected network is used to fully integrate user features $U_f$ and movie features $M_f$.The output is used as the predicted rating $R_p$, the model is optimized by continuously narrowing the loss between $R_p$ and the true rating $R_t$ [11]. Finally, the feature vector of the specified user and the feature vectors of all movies are used to calculate the predicted rating of each movie by the user, and the K movies with the highest predicted rating are selected to form a recommendation list.

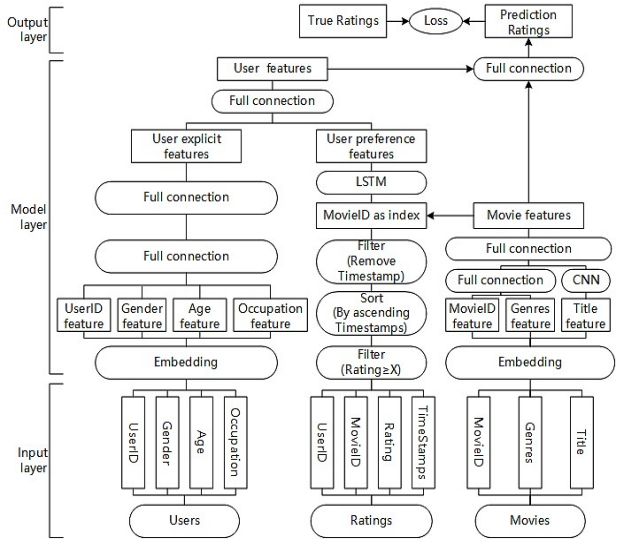


Figure 3. The combined recommendation model of LSTM and CNN.

## IV. Experiment and Result Analysis

### A. Data Set Introduction

This paper uses the MovieLens 1M data set, which was collected by the GroupLens project research group of the University of Minnesota in the United States [12]. The data set contains more than one million ratings from more than 6,000 users on nearly 4,000 movies. The data set consists of the following three files: user data file users.dat, movie data file movies.dat and ratings data file ratings.dat.

The original user data is shown in Table I, which contains a series of user information such as userID, gender, age, occupationID and zip-code.

TABLE I. Original User Data

| | UserID | Gender | Age | OccupationID | Zip-code |
|---|---|---|---|---|---|
| 1 | 11 | F | 25 | 1 | 04093 |
| 2 | 12 | M | 25 | 12 | 32793 |
| 3 | 13 | M | 45 | 1 | 93304 |
| 4 | 14 | M | 35 | 0 | 60126 |
| 5 | 15 | M | 25 | 7 | 22903 |
| 6 | 16 | F | 35 | 0 | 20670 |

The movies.dat file contains the movie ID, movie title, and movie genres. The original movies data is shown in Table II.

TABLE II. Original Movie Data

| | MovieID | Title | Genres |
|---|---|---|---|
| 1 | 11 | American President, The (1995) | Comedy\|Drama\|Roman |
| 2 | 12 | Dracula: Dead (1995) | Comedy\|Horror |
| 3 | 13 | Balto (1995) | Animation\|Children's |
| 4 | 14 | Nixon (1995) | Drama |
| 5 | 15 | Cutthroat Island(1995) | Action\|Adventure\|Roman |
| 6 | 16 | Casino (1995) | Drama\|Thriller |

The user ratings information is mainly stored in the ratings.dat file, which contains the userID, the movieID rated by the user, the ratings, and the timestamp. The original rating data is shown in Table III.

TABLE III. Original Ratings Data

| | UserID | MovieID | Rating | TimeStamps |
|---|---|---|---|---|
| 1 | 1 | 938 | 4 | 978301752 |
| 2 | 1 | 2398 | 4 | 978302281 |
| 3 | 1 | 2918 | 4 | 978302124 |
| 4 | 1 | 1035 | 5 | 978301753 |
| 5 | 1 | 2791 | 4 | 978302188 |
| 6 | 1 | 2687 | 3 | 978824268 |

In the ratings data, the user has five ratings for the movie. The higher the rating, the more the user likes the movie. The distribution of user ratings data are shown in Figure 4.

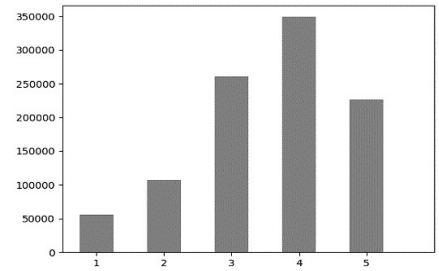


Figure 4. Distribution of user ratings data.

### B. Data Preprocessing

First, in the original user data, unused zip-code is discarded. In addition, the original user.dat gender field value uses English letters to indicate male or female. In order to facilitate subsequent processing, the English letters are replaced with 0 and 1. For the age field, according to its data characteristics, it is mapped to 7 levels of 0-6. Table IV shows the preprocessing results of user data.

For the movie data file movies.dat, the fields that need to be processed are the movie genres and movie title. For all movie genres, a text-to-number mapping dictionary is constructed, and then the genres of each movie are mapped into a list of numbers through the dictionary. Movie titles are treated in the same way. First, a mapping dictionary for each word in the title is constructed, and then each movie title is mapped into a list of numbers. For some field values with different lengths after conversion, corresponding padding operations will be performed. Table V shows the results of movie data preprocessing.

TABLE IV. USER DATA PREPROCESSING RESULTS

| | UserID | Gender | Age | OccupationID |
|---|---|---|---|---|
| 1 | 11 | 0 | 6 | 1 |
| 2 | 12 | 1 | 6 | 12 |
| 3 | 13 | 1 | 2 | 1 |
| 4 | 14 | 1 | 1 | 0 |
| 5 | 15 | 1 | 6 | 7 |
| 6 | 16 | 0 | 1 | 0 |

TABLE V. MOVIE DATA PREPROCESSING RESULTS

| | MovieID | Title | Genres |
|---|---|---|---|
| 1 | 11 | [2339,3285,289,3226…] | [6,4,0,1,1,1,1…] |
| 2 | 12 | [854,5069,1870,3675…] | [6,18,1,1,1,1,1…] |
| 3 | 13 | [1281,3226,3226,3226…] | [9,10,1,1,1,1,1…] |
| 4 | 14 | [1267,3226,3226,3226…] | [4,1,1,1,1,1,1…] |
| 5 | 15 | [4209,1971,3226,3226…] | [5,3,0,1,1,1,1…] |
| 6 | 16 | [3566,3226,3226,3226…] | [4,13,1,1,1,1,1…] |

### C. Feature Extraction

- User explicit feature extraction

After preprocessing, the embedding method is used to obtain the feature vectors of each field in the user data. Then, each feature will pass through a two-layer fully connected network to calculate the user explicit feature vector $U_{ef}$.

- Movie feature extraction

First, the pre-processed movie ID and movie genre information were obtained through embedding to obtain the feature vectors of each field. For the movie title, first convert each word of the movie title into embedding vectors, and then these vectors form the title embedding matrix. Next, the CNN convolution layer will be constructed, and multiple convolution kernels of different sizes are used to perform the convolution operation on the title embedding matrix. The output of the convolution layer is passed to the pooling layer and then to the maximum pooling operation. After the multi-step convolution and pooling operations, the result will be filtered in the dropout layer to obtain the feature vector of the movie title.

Next, the movie ID feature vector and the movie genres feature vector are through a fully connected network to obtain a fusion vector. Then, the fusion vector and the movie title feature vector are through the fully connected network again to obtain the movie feature vector $M_f$.

- User preference feature extraction

All movies rated by each user are considered a set U. Through the analysis of the ratings data, movies with a rating greater than or equal to 3 will be retained. The filtered data is sorted in ascending order according to timestamps, and the processed user rating set $U_s$ is obtained.

In order to extract the user preference feature $U_{pf}$ more completely, this paper will use the extracted movie feature $M_f$ to calculate $U_{pf}$. This paper uses two layers of LSTM. In the data import phase of the model, for each row of data in $U_s$, the movie ID is used as an index, and the corresponding movie feature vector is searched in $M_f$. This vector is passed into LSTM as the input data at the current moment. All the data of $U_s$ will be sequentially transferred to LSTM according to the above method for calculation, and finally the LSTM cell state will be output as the user preference feature $U_{pf}$. Use the fully connected network to fuse $U_{ef}$ and $U_{pf}$ to obtain the user feature $U_f$.

### D. Evaluation Indicators and Hyperparameter Settings

The model uses Mean Square Error (MSE) and Mean Absolute Error (MAE) for evaluation. MSE is the expected value of the square of the difference between the predicted value and the true value, and is often used to evaluate the degree of change in loss. The calculation method of MSE is shown in equation (10), where $R_t$ represents the true value, $\widehat{R_t}$ represents the model predicted value, and n represents the number of samples.

MAE represents the average value of the absolute error between the predicted value and the true value, which can accurately reflect the size of the actual prediction loss. The calculation method of MAE is shown in equation (11).

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\left(R_t - \hat{R}_t\right)^2 \tag{10}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|R_t - \hat{R}_t\right| \tag{11}$$

By adjusting the parameters of each part of the model, the loss is continuously optimized and the operating effect of the model is improved. In the end, the hyperparameter combinations that make the results better are obtained, as shown in Table VI.

TABLE VI. HYPERPARAMETER SETTINGS

| Hyperparameter settings | Value |
|---|---|
| epochs | 5 |
| batch_size | 256 |
| dropout_keep_prob | 0.5 |
| learning rate | 0.0001 |

### E. Experimental Results and Analysis

In the experiment, divide the data set into training set and test set at a ratio of 8:2. First use the training set for model training and optimize each part of the parameters. After the model training is complete, use the test set to verify the performance of the model.

Figure 5 shows the MSE loss on the training set and test set of the model proposed in this paper, and Figure 6 shows the corresponding MAE loss.

It can be obtained from Figure 5 that the MSE loss of the model has a downward trend on the training set and the test set as a whole, and finally stabilizes between 0.7-1.1. The MAE loss of the model shown in Figure 6. The MAE value on the training set converges within a short number of

training times. On the test set, the overall MAE has a downward trend, and finally remains between 0.65-0.80.

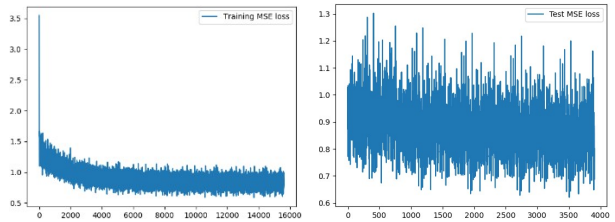


Figure 5. Combined recommendation model of LSTM & CNN MSE loss.

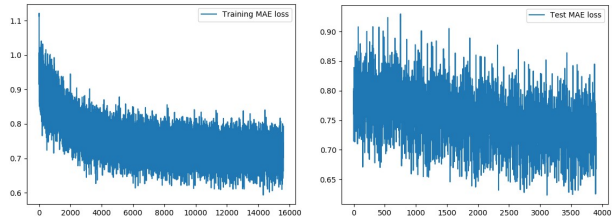


Figure 6. Combined recommendation model of LSTM & CNN MAE loss.

In order to further verify the performance of the combined recommendation model of LSTM and CNN, it is compared with the existing movie recommendation model. The comparison model used is divided into two categories: one is the traditional recommendation model, including user-based collaborative filtering (User-based CF) [13], item-based collaborative filtering (Item-based CF) and singular value decomposition (SVD) recommendation model [14]; the other is the deep learning-based recommendation model, including LSTM-based recommendation model [15], CNN-based recommendation model [16]. Comparison results are shown in Table VII.

TABLE VII. MSE LOSS AND MAE LOSS COMPARISON

| Model | MSE | MAE |
|---|---|---|
| LSTM & CNN | **0.876** | **0.751** |
| User-based CF | 0.945 | 0.781 |
| Item-based CF | 1.022 | 0.829 |
| SVD | 1.063 | 0.830 |
| LSTM | 0.920 | 1.273 |
| CNN | 0.935 | 0.979 |

It can be seen from Table VII that, compared with the traditional recommendation model and other deep learning-based recommendation models, the MSE and MAE values of the model proposed in this paper are smaller.

After the model training is completed, the user prediction rating for each movie is calculated using the feature vector of the specified user and the feature vectors of all movies, and the recommendation list of the five movies with the highest predicted rating is recommended to the user. The recommended results are shown in figure 7, where each line of information is in turn the movie ID, movie title, movie genres, and prediction rating (round up to 2 decimal places).

```
[3377, 'Hangmen Also Die (1943)', 'Drama|War',4.28],
[556, 'War Room, The (1993)', 'Documentary',4.27],
[3378, 'Ogre, The (Der Unhold) (1996)', 'Drama',4.03],
[3797, 'In Crowd, The (2000)', 'Thriller',3.96],
[3461, 'Lord of the Flies (1963)', 'Adventure|Drama|Thriller',3.88]
```

Figure 7. Recommendation Results.

## V. SUMMARY

This paper focuses on movie recommendation and proposes a combined recommendation model of LSTM and CNN. The model combines CNN to fully mine the local information of movie data, and uses LSTM to capture the context of user ratings. Through feature fusion, the prediction ratings are calculated and a recommendation list is formed. Through experiments, the model proposed in this paper is better in the accuracy of rating prediction, and can recommend movies that are more in line with users' preferences.

## ACKNOWLEDGMENT

This work is funded by the program of Qinghai Province Applied Basic Research (No.2018-ZJ-787).

## REFERENCES

[1] Haibo Liu. Resource recommendation via user tagging behavior analysis[J]. Cluster Computing, 2019, 22(4):1-10.

[2] Huang Li Wei, Jiang Hei Wu, Wu Mamoru , et al. Basic Depth Science Investigative Research [J]. Calculator, 2018,41 (07): 1619-1647.

[3] Gong Y , Zhang Q . Hashtag Recommendation Using Attention-Based Convolutional Neural Network[C]// International Joint Conference on Artificial Intelligence. AAAI Press, 2016.

[4] Jeffrey Lund , Yiu-Kai Ng .Movie Recommendations Using the Deep Learning Approach[C]// 2018 IEEE International Conference on Information Reuse and Integration (IRI).2018.

[5] Sak, Haşim, Senior A , Beaufays, Françoise. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition[J]. Computer Science, 2013.

[6] Zhou Feiyan, Jin Linpeng, Dong Jun. A review of convolutional neural network research [J]. Chinese Journal of Computers, 2017, 40 (06): 1229-1251.

[7] Li Yandong, Hao Zongbo, Lei Hang. A review of convolutional neural network research [J]. Computer Applications, 2016, 36 (09): 2508-2515 + 2565.

[8] Jin R , Lu L , Lee J , et al. Multi-representational convolutional neural networks for text classification[J]. Computational Intelligence, 2019(11).

[9] Zhang Yifan, Huang Yixiang, Wang Kaizheng. Parallel combination model of LSTM and CNN for arrhythmia recognition [J]. Journal of Harbin Institute of Technology, 2019, 51 (10): 76-82.

[10] Senyurek V Y , Imtiaz M H , Belsare P , et al. A CNN-LSTM neural network for recognition of puffing in smoking episodes using wearable sensors[J]. Biomedical Engineering Letters, 2020(1).

[11] Chenbin Li, Guohua Zhan, Zhihua Li. News Text Classification Based on Improved Bi-LSTM-CNN[C]// 2018 9th International Conference on Information Technology in Medicine and Education (ITME). 2018.

[12] MoviesLens[OL].https://grouplens.org/datasets/MovieLens/.

[13] Jiang S, Zhang L, Zhang Z. New Collaborative Filtering Algorithm Based on Relative Similarity[J]. 2016.

[14] Koren Y , Bell R , Volinsky C . Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8):30-37.

[15] Devooght R , Bersini H . Collaborative Filtering with Recurrent Neural Networks[J]. 2016.

[16] Ruan Wenjun, Hu Xiaolong, Li Lihua. Movie recommendation algorithm based on deep learning [J]. Journal of Hubei University (Natural Science Edition), 2020, 42 (02): 136-141.