Received July 21, 2019, accepted August 1, 2019, date of publication August 15, 2019, date of current version September 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935224

Collaborative Filtering Recommendation Algorithm for Heterogeneous Data Mining in the Internet of Things

YING GAO[1,2] AND LINGXI RAN[3]
[1]Zhou Enlai School of Government, Nankai University, Tianjin 300000, China
[2]College of Life Sciences, Inner Mongolia University for Nationalities, Tongliao 028043, China
[3]Electrical Engineering Department, Shandong University, Shandong 120003, China

Corresponding author: Lingxi Ran (rlxqingdao@163.com)

This work was supported in part by the Inner Mongolia Autonomous Region of Science and Technology Innovation Guiding Project under Grant KCBJ2018028, in part by the Inner Mongolia University for Nationalities Scientific Research Projects under Grant NMDYB1730, in part by the College Scientific Research Project of the Inner Mongolia Autonomous Region under Grant NJZY18159, in part by the Natural Science Foundation of the Inner Mongolia Autonomous Region of China under Grant 2018MS05025, and in part by the Program for Young Talents of Science and Technology in the Universities of Inner Mongolia Autonomous Region under Grant NJYT-19-B15.

ABSTRACT With the popularization of Internet of Things (IOT) technology, a large number of multi-source heterogeneous data are constantly generated and collected by cloud platforms, which indicates that the problem of large data in IOT has become increasingly prominent, especially for massive tags and information in IOT which is urgent to use appropriate data mining algorithms to mine the value of these data. A collaborative filtering recommendation algorithm based on multi-information source fusion (CFR-MIF) is proposed where a feature vector and time weight function are introduced to improve the accuracy of top-N recommendation. It can conveniently and effectively process the IoT data, and furthermore integrate, manage and store the massive data collected from different industries and data formats. Besides, It also provides data mining services in the whole IoT realizing prediction and decision-making, which reverses control these sensor networks, so as to control the movement and development process of objective in the Internet of Things. The experimental results based on DeviceLens 1M data set show that the proposed algorithm greatly improves the accuracy of recommendation results, recall rate and F1 value compared with other advanced algorithms.

INDEX TERMS Collaborative filtering recommendation, information fusion, time factor, IOT.

I. INTRODUCTION

With the development of Internet of Things technology, the number of devices on IOT is growing exponentially. It is shown in fig 1 that the information which can be used for data mining by enterprises has further exploded, reaching astronomical figures. IOT cloud platform is a customized platform just for IOT, which differs much from the ordinary Internet in the number of devices, the total amount of data, the type of protocol, the access mode and so on. Based on the recent Internet and communication technology to build the IOT cloud platform, the construction of the platform needs to be considered from the aspects of communication protocol, user management, equipment management, data packet analysis, data storage, large data application display, etc. The figure 2 shows an overall architecture of the typical IOT platforms. By fully mining the multi-source heterogeneous data generated by the Internet of Things, it can help enterprise decision makers to better achieve crowd portraits and accurately know user needs and satisfaction. Nowadays, many Internet of Things enterprises are actively exploring in the field of data mining and utilization. The research object of this paper chooses a classical problem in the field of data mining: recommendation system as shown in figure 1.

In today's era of Internet information overload, users often need to spend a lot of time and energy to select an item in the face of massive amounts of information [1], [2]. In order to solve the problem of information overload, there are currently two major research directions: search engine and recommendation system [3]. The search engine solves the

The associate editor coordinating the review of this article and approving it for publication was Francesco Piccialli.

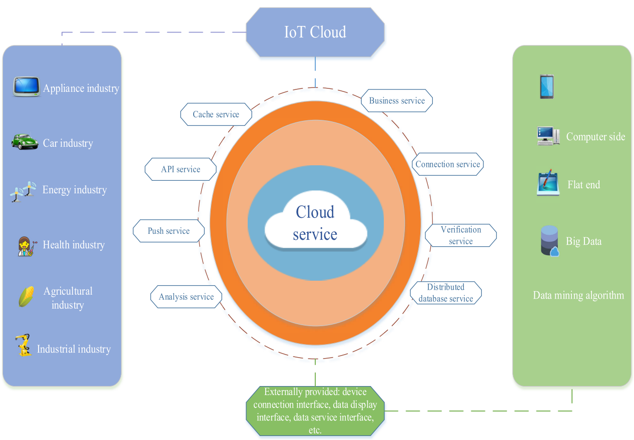

FIGURE 1. **Panorama of the internet platform.**

target-specific information retrieval, but many times the user is not aware of their own needs or can not accurately describe their needs, and so, the search engine can not solve the user's demand problem well. The recommendation system can use data mining and other technologies to provide users with personalized information recommendations [4].

At present, the mainstream recommendation systems are mainly divided into four categories, that is content-based recommendation, collaborative filtering recommendation, knowledge-based recommendation, and combination recommendation [5], [6]. Among them, the collaborative filtering algorithm can generate recommendations only based on the rating characteristics of similar users or projects, and can discover the potential information needs of the users [7], [8], not requiring the attribute information of the user or the project. And so, it has strong adaptability in different applications and is widely used [9].

Compared to the content-based recommendation algorithm, although the collaborative filtering recommendation algorithm does not depend on the feature information of the project and is also not limited to the limitations of the content analysis technology, it is limited by the problem of data sparsity [4], [10].

Sparsity has a direct impact on the quality of recommendation system, which has attracted great attention from academia and application. At present, there are many ways to solve the sparsity problem, such as simple filling, clustering, dimensionality reduction, content-based filtering, etc.

Breese [11] improves the resolution of similarity by adding some default scores. Simple filling method can alleviate the problem of data sparsity to a certain extent. But when the number of users and items is large, it need to fill all default values and the recommended calculation is relatively large. Generally, it is suitable for small-scale database. In addition, there will be some differences in the user's evaluation of the items that have not been overrated. This method uses a unified numerical value to fill in, without considering the user's interest differences.

Clustering method improves the accuracy of prediction by using the scoring information of similar groups. Koohi and Kiani [12] employed a subspace clustering approach to solve data sparsity by finding the neighbor user. The result shows that it is efficient in dealing with sparse data. Combining collaborative filtering (CF) technique and fuzzy c-means (FCM) clustering algorithm, Verma *et al.* [13] set up a recommendation system. Aimed at the issues of sparsity and scalabilities, Nitin and Fan [14] proposed a hybrid collaborative filtering method. And some personalized recommendations are recommend to solve them. Based on fuzzy C-means clustering algorithm, Koohi and Kiani [15] set up a collaborative filtering recommendation system. The K-means and SOM clustering approaches have been evaluated. The precision and recall are improved. But cluster can not reflect the differences of user preferences so that the accuracy of recommendation results has not been significantly improved.

Although dimensionality reduction method can reduce the scale and sparseness of user-item scoring matrix to a certain extent, it also loses some users' scoring data. It is difficult to guarantee the effect of dimensionality reduction when the dimensionality of item space is very high.

Collaborative filtering recommendation algorithms can be divided into user-based collaborative filtering (UCF) and Item-based Collaborative Filtering (ICF). UCF finds the nearest neighbor set of the target user according to the similarity between users, and then determines the recommendation result of the target user according to the user's rating in the set.

Inserting a user reduction procedure in traditional UCF, Zhang *et al.* [16] proposed a novel approach using covering-based rough set theory, which improves the accuracy and coverage of recommender systems. Combining cluster analysis with data fitting to extract user interests, Liu [17] has proposed a novel user interest model to compute the user interest degree. Based on this model, a collaborative filtering approach is used to rank the candidate resources.

ICF is the result of recommendation list by analyzing the similarity between projects, and ultimately taking the project with better evaluation by target users as the result of recommendation list. Pirasteh *et al.* [18] improve the accuracy of the recommendation by converting the symmetry similarity obtained by the traditional calculation method to asymmetric similarity and weighting to consider the user's score frequency for each score, when calculating user similarity and taking into account the different number of different user-rated items.

Collaborative filtering may bring Matthew effect. So some common improvements are proposed. Item-based improvement mainly includes item centralization and user centralization, that is, first subtracting the average of items and user scores, and then calculating similarity [18].

User-based collaborative filtering mainly improves the degree of users' preference for items, such as punishing the degree of preference for hot items, increasing the time attenuation of preference, and so on. Xia *et al.* [19] introduced the concept of time decay and proposed a method to calculate the similarity of items by incorporating the time decay function, which improved the accuracy of project-based collaborative

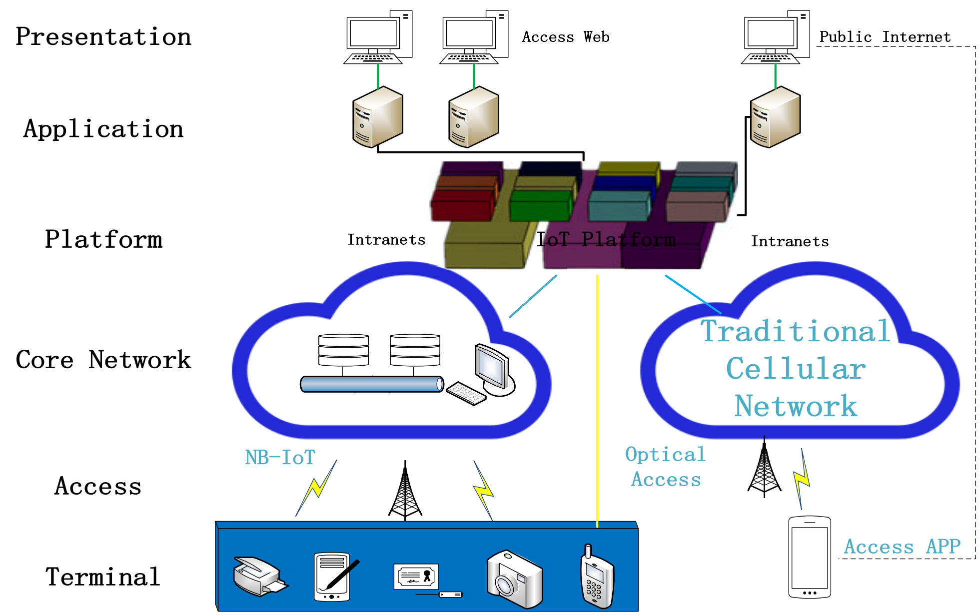

FIGURE 2. **Overall architecture of the IOT platform.**

filtering recommendations to some extent. Based on the traditional collaborative filtering algorithm, Patra *et al.* [20] added a time weight function to improve the accuracy of collaborative recommendation based on changes of group user preferences over time. Zhang *et al.* [21] divided the user's historical score into several periods, analyzed the user's interest distribution in each period, and then set a time window to find the user's recent interest. These above algorithms improve the recommendation accuracy to a certain extent after the integration of time factors. However, they do not dig deeply into the user-item scoring matrix and do not fully consider the user's scoring characteristics. And so, there is still room for improvement in the accuracy of the recommendation.

Therefore, aims at the data sparseness problem, fully mines the score matrix information, uses the user preference model, considers the asymmetrical influence degree between users, and constructs a time weight function, a collaborative filtering recommendation algorithm based on multiple information sources fusion (CFR-MIF) is proposed in this paper as shown in fig 3. The experimental results on the DeviceLens1M datasets show that, the proposed algorithm has greatly improved the accuracy, recall rate and F1 value of the recommendation results.

II. DESCRIPTION OF THE COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM

The core of the recommendation system is m user sets and n item sets. The following is an example of user-based collaborative filtering [22] to introduce the recommendation process.

Step 1: build the scoring matrix. The user - level scoring of item sets is converted into scoring matrix S (m, n). Generally, the score ranges from 0 to 5, and the higher the score, the more satisfied the users are.

Step 2: find neighbor users by scoring matrix. In this paper, Pearson correlation coefficient is selected to calculate the similarity degree. The formula is shown in (1)

$$clo(p,q) = \frac{\sum_{i \in \boldsymbol{I}_{pq}} (t_{pi} - \bar{t}_p)(t_{qi} - \bar{t}_q)}{\sqrt{\sum_{i \in \boldsymbol{I}_{pq}} (t_{pi} - \bar{t}_p)^2 (t_{qi} - \bar{t}_q)^2}} \tag{1}$$

where, $\boldsymbol{I}_{pq}$ refers to the item set with the same score of user p and q, t_{pi} refers to the score of user p on project i , and t_{qi} refers to the score of user q on project i.

Step 3: provide reasonable recommendations to target users.

(1) determine candidate projects, and then calculate the predicted scores of target users for candidate projects. Candidate projects refer to those projects that are rated by neighboring users q but not by target users. The calculation method is shown in equation (2) [23]:

$$t_{pi} = \frac{\sum_{p \in P_m} clo(p,q) \times (t_{pi} - \bar{t}_p)}{\sum_{p \in P_m} clo(p,q)} + \bar{t}_q \tag{2}$$

where, $\bar{t}_p$ refers to the average score obtained by processing target user data, and $\bar{t}_q$ refers to the average score obtained by processing neighbor user data. P_m represents the set of nearest neighbors of the target user.

(2) recommend the most reasonable projects to target users. The above can be calculated to obtain the predicted scores of the target users for the project, and the first few projects with the highest scores are recommended to the target users. It is worth mentioning that when it comes to recommending reasonable projects to target customers, the predicted score is key, but the degree of interest of target users in the project is more important. Therefore, equation (2) can be simplified to

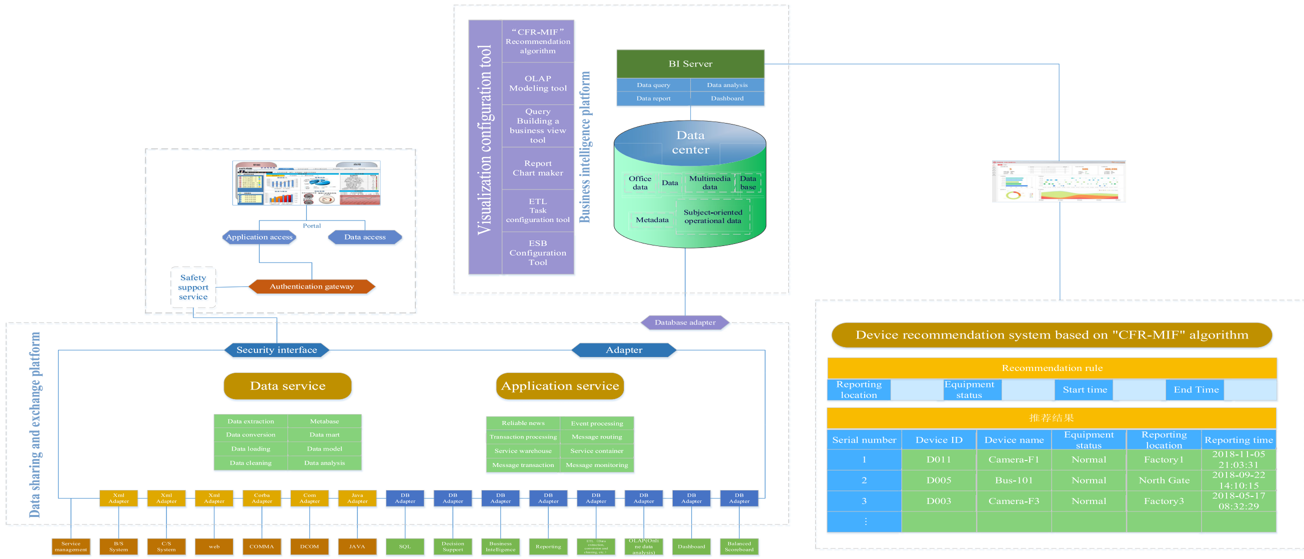

FIGURE 3. Equipment recommendation system based on CFR-MIF algorithms.

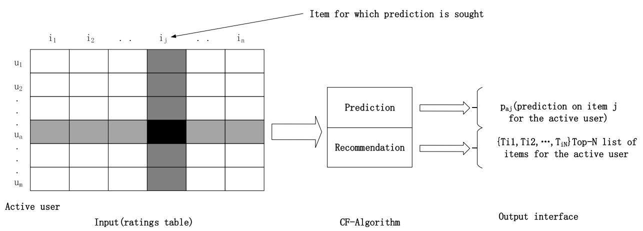

FIGURE 4. Classical collaborative filtering recommendation algorithm.

obtain equation (3).

$$t_{pi} = \sum\nolimits_{p \in P_m} clo(p, q) \times (t_{pi} - \bar{t}_p) + \bar{t}_q \tag{3}$$

where, t_{pi} efers to the degree of interest of target users in candidate projects, which is different from the meaning referred to by t_{pi} in equation (2). Other symbols have the same meanings as those expressed above and will not be repeated here. Figure 4 shows the degree of user interest in the candidate project.

III. THE PROPOSED CFR-MIF ALGORITHM

The steps of collaborative filtering recommendation algorithm for heterogeneous data mining in the Internet of things are as follows: step 1: build a user preference model, and convert the explicit score of the target customer into the implicit score as far as possible; Step 2: calculate the degree of asymmetric influence between users and eliminate the interference of special data as much as possible; Step 3: build the time weight function to obtain the preference degree of target users to the project at different moments; In step 4, target customers are provided with the items with the highest preference scores obtained.

In order to achieve better expression effect, this paper selects 4 users' scores of 8 items as data and obtains table 1. The scoring standard adopts a 5-point system,

TABLE 1. Example of user-equipment score matrix.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
P_1	2	2	1	2	2	3	5	5
P_2	3	1	3	3	4	4	2	5
P_3	1	0	0	4	0	0	0	1
P_4	2	2	1	1	3	3	4	0

5 represents the most satisfied, 0 represents the user does not participate in the scoring, and so on.

A. USER PREFERENCE MODEL

It is well known that users have different rating standards [13], [24]. For example, user p is used to giving 5 points to the highest score and user q is used to giving 3 points to the average. If both p and q give 5 points to a certain project, it can be clearly judged that p enjoys a certain project more than q. Therefore, it is necessary to establish a user preference model to exclude the influence of different users' scoring standards as much as possible. You might as well set up A grading level, combined with the above 5-point system, you can set the grading level as $\{B_1, B_2, \cdots, B_k\}$, $B_i \rangle B_j$, and so on. Through equation (4), the user's preference score for B_i can be calculated [13].

$$qus(B_i) = a \cdot \frac{qus\,|B_i|}{S_p} + b \cdot \sum\nolimits_{B_j \in \{B_1, B_2, \cdots, B_k\}} \frac{qus\,\left||B_j|\right|}{S_p} \tag{4}$$

where, S_P refers to the number of ratings given by user p, $qus\,|B_i|$ refers to the number of ratings given by user B_i, and $qus\,\left||B_j|\right|$ refers to the number of ratings given by user B_j. The values of parameters a and b can be determined by reference [13], [25], $a = 2, b = 0.8$.

According to equation (4), users' preference score for category D_8 can be calculated. Based on the data in table 1,

TABLE 2. The preference score of user u_1, u_2 for each scoring category.

	B_1	B_2	B_3	B_4	B_5
P_1	0.13	0.23	0.35	0.45	0.81
P_2	0.15	0.10	0.32	0.69	0.93

TABLE 3. User- equipment preference score matrix of user u_1, u_2.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
P_1	0.22	0.13	0.33	0.35	0.55	0.80	0.81	0.80
P_2	0.15	0.20	0.28	0.38	0.78	0.58	0.68	0.93

table 2 can be obtained. The data in table 3 refer to the data converted into the preference score matrix. According to the analysis, in table 1, both p1 and p2 rate D_8 at 5. If we look at this data alone, we can only assume that two users have the same degree of preference for a. however, according to the data in table 3 calculated by the user preference model, we can find that the two users have different degrees of preference for D_8. Obviously, the construction of user preference model has certain effect on excluding the influence of user rating criteria.

After building the user preference model, when calculating score prediction for users, it is necessary to convert equation (3) appropriately. After conversion, equation (5) is obtained. The function of equation (5) is to calculate the updated version of user preference score [20], [26].

$$t_{pi} = \sum\nolimits_{q \in P_m} clo(p, q) \times qus(t_{qi}) \tag{5}$$

where, $qus(t_{qi})$ refers to the preference score of neighboring user q for item I, and the other symbols refer to the same meaning as above, which is not repeated here.

B. ASYMMETRICAL INFLUENCE BETWEEN USERS

Due to the large differences among individual users, the degree of mutual influence among users also varies greatly [27]. In order to quantify the differences in the degree of mutual influence among different users, this paper introduces the degree of asymmetric influence among users. The calculation method of the degree of asymmetric influence is shown in equation (6) [12]:

$$influence(p, q) = \frac{1}{exp(\frac{|\boldsymbol{I}_p \cup \boldsymbol{I}_q|}{|\boldsymbol{I}_p|} - 1)} \tag{6}$$

According to equation (6), the influence degree of user p1 on user p2 is 0.632, and that of user p2 on user p1 is 0.283, $influence(p1, p2) > influence(p2, p1)$. Analysis and calculation results show that the degree of mutual influence between users is quite different.

C. TIME WEIGHT FUNCTION

We know that users are emotional, and their preferences in the near stage may be different from those in the past. Similarly, users' preferences for project i in the past may be different from those for project i now. In order to reduce the influence of time on scoring, it is necessary to introduce time weighting function. Time weight function is shown in equation (7) [18], [28]:

$$T(time(t_{pi})) = \frac{1}{exp[k \cdot (t_0 - time(t_{pi}))]} \tag{7}$$

Among them, $time(t_{pi})$ refers to the time when the user gave score to project I in the past, t_0 refers to the time when the staff sorted out the user's score, and k is a parameter, but this parameter is related to time to some extent. The larger the score interval is, the greater its value will be. Through the formula, we can know that the greater the interval between the scores, the lower the credibility of the given scores.

D. PREDICTIVE PREFERENCE SCORE

Users' scoring standards are different, and time has an impact on the scoring credibility. The above two factors have an impact on the user's prediction preference score, so the final prediction score of users is the result calculated by (8).

$$t_{pi} = \sum\nolimits_{p \in P_M} clo(p, q) \times qus\left(t_{pi}\right) \times influence(p, q) \times T(time(t_{pi})) \tag{8}$$

where, $clo(p, q)$ is calculated by equation (1), $qus\left(t_{pi}\right)$ by equation (4), $\inf luence(p, q)$ by equation (6), and $T(time(t_{pi}))$ by equation (7).

E. DETAILED STEPS OF THE PROPOSED CFR-MIF ALGORITHM

The detailed steps of the proposed collaborative filtering recommendation algorithm based on multi-source fusion are as follows:

Input: Target location, location equipment rating matrix, number of location neighbors and previous N recommended device numbers..

Output: Set of devices including devices recommended for use at target locations.

Step1: Calculating each user's preference score for each rating category {B1, B2, B3, B4, B5} according to formula (4) based on the user equipment rating matrix S (m, n) and then creating the equipment rating matrix and the score recording matrix.

Step 2: Averaging all elements by mean function based on the calculated scoring matrix where score of each user is equal to the original score subtracting average score, where M represents the number of devices and N represents the number of users.

Step 3: Setting as candidate sets for all devices that users are rated but the target users are not rated to display scalar information, and training models whose all summaries are saved in disk for output.

Step 4: Adding user content matrix and user preference matrix based on the model trained in the previous step where matrix multiplication transformation is used to calculate the square root of each element and evaluate the model.

Step 5: Calculating predictive preference scores for target users of all candidate devices using the evaluation model and

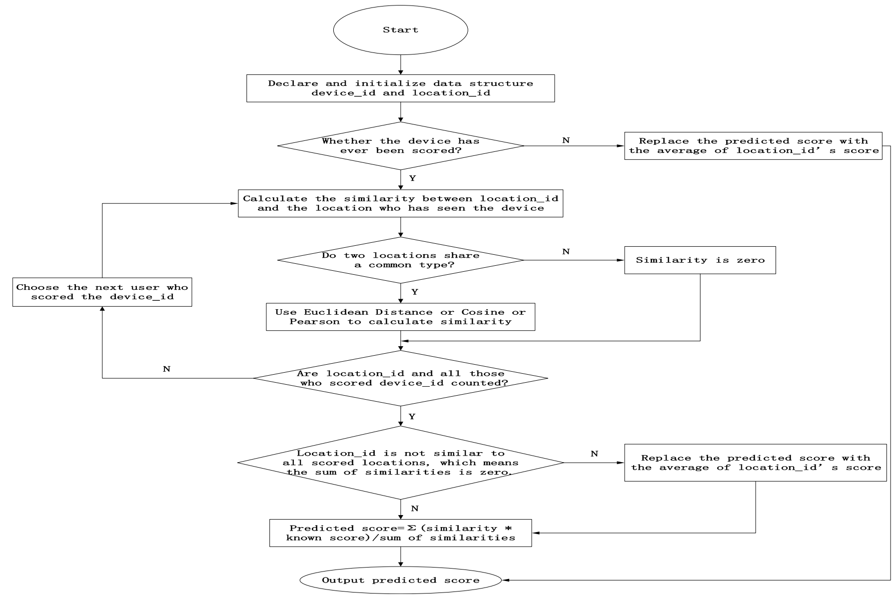

FIGURE 5. Flow chart of CFR-MIF algorithm.

generating Top - N recommendation lists by getting the highest score device from the target users,as shown in figure 5.

IV. EXPERIMENTAL CONSTRUCTION AND ASSESSMENT METHODS

A. DATA SETS

With the Internet of Things (IOT) coming to the ground and being applied in the industry, the high speed of data generation, the various types and the huge volume of data will overwhelmingly bring more pressure and challenges to the existing data mining and data processing methods. The IoT data can be divided into static data and dynamic data. Static data are mostly label data and address data, such as data generated by RFID, which are mostly stored in structured and relational databases. Dynamic data are time-series data and the characteristics of dynamic IoT data are that each data has a one-to-one relationship with time which is particularly important in data process. This kind of data storage is usually stored in a sequential database. In this experiment, we extracted some equipment information from production as experimental data, including 723321 ratings of 15231 equipment from 2,213 locations where the state rangs from 1 to 5.

In order to verify the performance improvement of the recommendation algorithm, the original data set is adjusted as follows: the original data set is extracted as a sub-data set according to different time intervals, where the first 80% is used as a training set and left 20% is set as a test set.

B. EVALUATION INDEX

The current major evaluation indicators, Precision rate and Recall rate [29], is used to evaluate the performance of the recommended algorithm in this experiments. the accuracy rate refers to the ratio of the number of correctly recommended items to the number of all recommended items. It indicates the probability that the user is interested in the system recommendation item and measures the recommendation effect to the user. The recall rate indicates the probability that a user's favorite item is recommended, and it is defined as the ratio of the user's favorite item in the recommendation list to all the items the user likes in the system. The recall rate is mainly measured at the system level.

The accuracy rate of the Recommended system is:

$$\text{Accuracy} = \frac{\sum_{\text{p}\in\text{P}} |\text{C(p)}|}{\sum_{\text{p}\in\text{P}} |\text{C(p)} \cup \text{D(p)}|} \tag{9}$$

The recall rate of the Recommended system is:

$$\text{Rec} = \frac{\sum_{\text{p}\in\text{P}} |\text{D(p)}|}{\sum_{\text{u}\in\text{U}} |\text{C(p)} \cup \text{D(p)}|} \tag{10}$$

where, C(p) represents the set of user-generated recommendation items, D(p) represents the set of items scored by users in the test set, and P represents the set of all users in the test set.

In order to evaluate the advantages and disadvantages of the proposed algorithm more accurately, and at the same time to consider the accuracy and recall rate, this index is also used as the evaluation index of the performance of the proposed algorithm. The calculation formula is as follows:

$$\text{T}_1 = \frac{\text{Accuracy} + \text{Rec}}{3 \times \text{Accuracy} \times \text{Rec}} \tag{11}$$

The smaller the T_1 value, the better the overall performance of the recommended algorithm.

V. ANALYSIS OF THE EXPERIMENT RESULTS

To verify the accuracy of the proposed collaborative filtering recommendation Realgorithm CFR-MI, we calculate the similarity between different devices offline in advance and

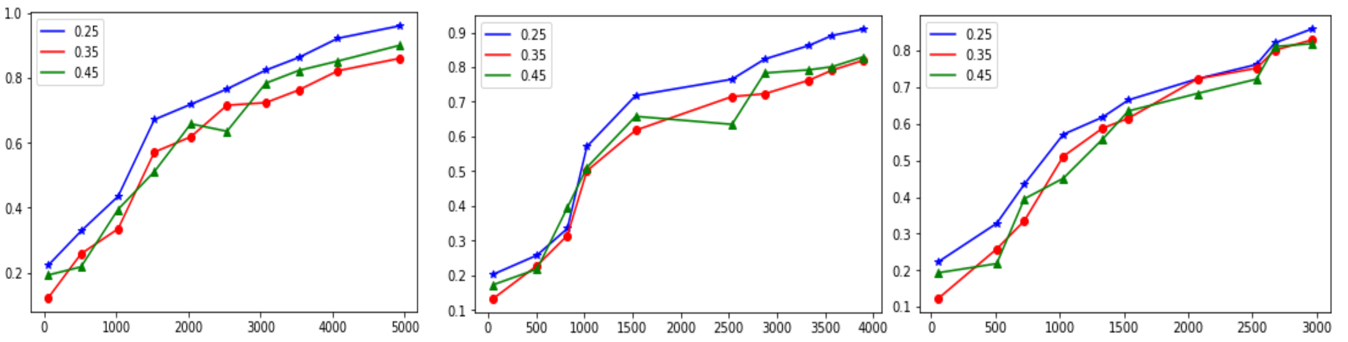

FIGURE 6. Accuracy of CFR-MIF algorithm on different data samples.

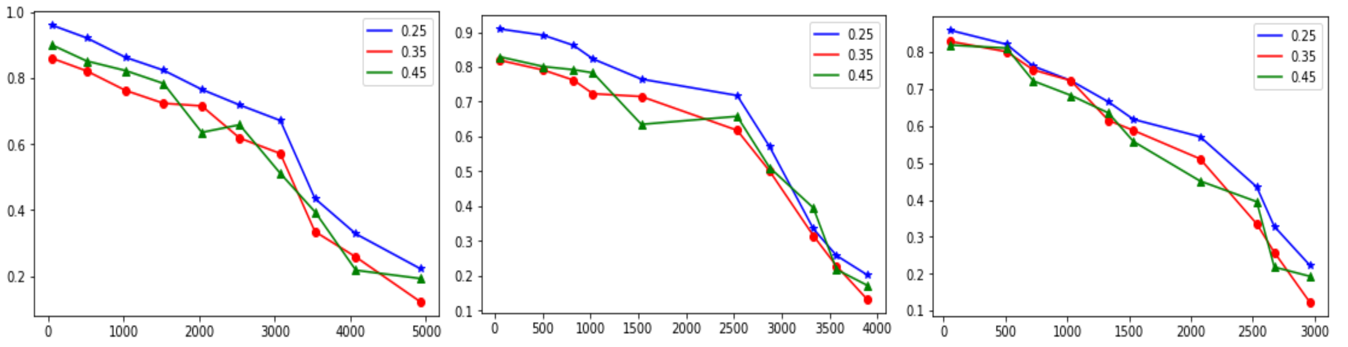

FIGURE 7. Loss rate of CFR-MIF algorithms on different orthodox distributions.

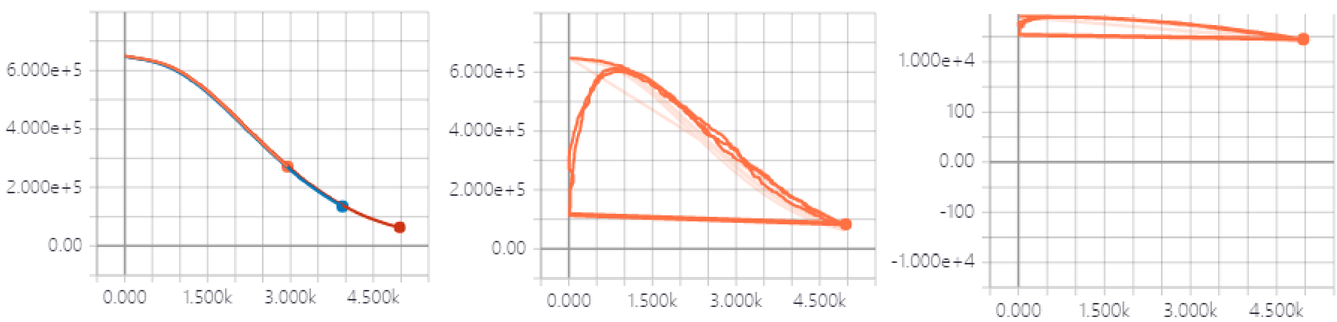

FIGURE 8. Loss rate of CFR-MIF under different data samples, standard deviation and shrinkage algorithms.

save it as a model, because the direct similarity of devices is relatively fixed. By using this model, the influence of time weight of devices and adjacent data are reduced to improve the recommendation quality. Based on the collaborative filtering recommendation algorithm of multi-information source fusion, we introduce a feature vector and time weight function to improve the accuracy of top-N recommendation, and we experimentalized on DeviceLens 1M data set.

Firstly, the performance and loss rate of the proposed CFR-MIF algorithm on different training data sets are tested. Fig. 6 shows the accuracy of different data sets.

As can be seen from Figure 6, the accuracy index of the proposed CFR-MIF algorithm varies with the amount of data in DeviceLens 1M data set. Further analysis of the experimental results also shows that when the time weight is the recommended result, the recommended result is the best, which represents all the scoring time periods and training sets. For example, in DeviceLens 1M data set, when the number of training set samples reaches 5000, the data presents a normal distribution. Therefore, in all subsequent experiments, for experiments using DeviceLens 1M data set, the data volume sampling is set to 5000.

Figure 7 shows the accuracy of the recommended CFR-MIF algorithm. When the number of recommended devices is determined, the function is used to extract the specified number of values from the values obeying the specified orthogonal distribution. By adjusting the standard deviation of the normal distribution, the standard deviation is set to 0.35.

Under the experimental conclusions of Figures 6 and 7, we will further evaluate the experimental model. In the evaluation, we recommend the top 10 devices with the highest

```
Where would you like to recommend? Please enter the location number: location011
==========The top 10 devices recommended for the location are: ==========
Score: 5.56, Device name: Camera-F1
Score: 5.53, Device name: Bus-101
Score: 5.44, Device name: Camera-F3
Score: 5.36, Device name: Camera-F2
Score: 5.26, Device name: Bus-105
Score: 5.22, Device name: Device002
Score: 5.22, Device name: Device005
Score: 5.21, Device name: Camera-F5
Score: 5.12, Device name: Camera-F8
Score: 5.01, Device name: Bus-103
```

FIGURE 9. Top 10 devices with the highest recommended score based on location number.

score through the input location number. The recommended results are shown in Figure 9.

The experimental results show that the integration of multiple information sources, such as users' rating time and the asymmetry between users' preference models, can improve the performance of the whole algorithm and recommendation quality. By using this model, the time weights of devices and problems of adjacent data can be reduced to improve recommendation quality.

VI. CONCLUSION

Aiming at the problem of multi-source heterogeneous data mining often encountered in Internet of Things application system, this paper proposes a collaborative filtering recommendation algorithm based on multi-information source fusion (CFR-MIF). The location preference model is introduced and predicted by making full use of the location equipment characteristics at the base of traditional location-based collaborative filtering recommendation algorithm. Besides, a time weighting function is introduced considering the asymmetric effect between locations. The experimental results based on DeviceLens 1M data set show that the proposed algorithm can effectively improve the accuracy, recall rate and F1 value of the recommendation system. The idea of considering time factor and location preference model is universal and can be applied to other recommendation algorithms. The disadvantage of this algorithm is that it does not consider the cold-start problem. If the cold start problem is considered, the recommendation system built in this paper will be more perfect.

REFERENCES

[1] S. Khusro, Z. Ali, and I. Ullah, "Recommender systems: Issues, challenges, and research opportunities," in *Information Science and Applications*. Singapore: Springer, 2016.

[2] H. Naji and G. Al-Sultany, "Collaborative filtering recommendation using personalized page rank algorithm with new personalized parameters," *J. Eng. Appl. Sci.*, vol. 12, no. 16, pp. 4108–4112, 2017.

[3] C.-L. Liao and S.-J. Lee, "A clustering based approach to improving the efficiency of collaborative filtering recommendation," *Electron. Commerce Res. Appl.*, vol. 18, pp. 1–9, Jul./Aug. 2016.

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[5] A. Karatzoglou, B. Hidasi, D. Tikk, O. Sar-Shalom, H. Roitman, B. Shapira, and L. Rokach, "RecSys'16 workshop on deep learning for recommender systems (DLRS)," in *Proc. ACM Conf. Recommender Syst.*, 2016, pp. 415–416.

[6] X. W. Meng, S. D. Liu, Y. J. Zhang, and X. Hu, "Research on social recommender systems," *J. Softw.*, vol. 26, no. 6, pp. 1356–1372, 2015.

[7] S. Vairachilai, M. K. Kavithadevi, and M. Raja, "Alleviating the cold start problem in recommender systems based on modularity maximization community detection algorithm," *Circuits Syst.*, vol. 7, no. 8, pp. 1268–1279, 2016.

[8] L. Jp, Y. Wang, and Y. Fh, "An improved collaborative filtering recommendation algorithm," *Comput. Modernization*, vol. 32, no. 9, pp. 204–208, 2017.

[9] X. Ma, H. Lu, Z. Gan, and J. Zeng, "An explicit trust and distrust clustering based collaborative filtering recommendation approach," *Electron. Commerce Res. Appl.*, vol. 25, pp. 29–39, Sep./Oct. 2017.

[10] X. Liu, "An improved clustering-based collaborative filtering recommendation algorithm," *Cluster Comput.*, vol. 20, no. 2, pp. 1281–1288, 2017.

[11] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, 1998, pp. 43–52.

[12] H. Koohi and K. Kiani, "A new method to find neighbor users that improves the performance of collaborative filtering," *Expert Syst. Appl.*, vol. 83, pp. 30–39, Oct. 2017.

[13] S. K. Verma, N. Mittal, and B. Agarwal, "Hybrid recommender system based on fuzzy clustering and collaborative filtering," in *Proc. 4th Int. Conf. Comput. Commun. Technol. (ICCCT)*, Allahabad, India, Sep. 2013, pp. 116–120.

[14] N. P. Kumar and Z. Fan, "Hybrid user-item based collaborative filtering," *Procedia Comput. Sci.*, vol. 60, pp. 1453–1461, Apr. 2015.

[15] H. Koohi and K. Kiani, "User based collaborative filtering using fuzzy C-means," *Measurement*, vol. 91, pp. 134–139, Sep. 2016.

[16] Z. Zhang, Y. Kudo, and T. Murai, "Neighbor selection for user-based collaborative filtering using covering-based rough sets," *Ann. Oper. Res.*, vol. 256, pp. 359–374, Sep. 2017.

[17] H. Liu, "Resource recommendation via user tagging behavior analysis," *Cluster Comput.*, vol. 22, pp. 885–894, Jan. 2019.

[18] P. Pirasteh, D. Hwang, and J. E. Jung, "Weighted similarity schemes for high scalability in user-based collaborative filtering," *Mobile Netw. Appl.*, vol. 20, no. 4, pp. 497–507, 2014.

[19] C. Xia, X. Jiang, S. Liu, Z. Luo, and Z. Yu, "Dynamic item-based recommendation algorithm with time decay," in *Proc. 6th Int. Conf. Natural Comput. (ICNC)*, Yantai, China, Aug. 2010, pp. 242–247.

[20] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data," *Knowl.-Based Syst.*, vol. 82, pp. 163–177, Jul. 2015.

[21] H. Zhang, W. Ni, X. Li, and Y. Yang, "Modeling the heterogeneous duration of user interest in time-dependent recommendation: A hidden semi-Markov approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 2, pp. 177–194, Feb. 2018.

[22] B. Karahodza, H. Supic, and D. Donko, "An approach to design of time-aware recommender system based on changes in group user's preferences," in *Proc. 10th Int. Symp. Telecommun.*, Oct. 2014, pp. 1–4.

[23] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 39–46.

[24] D. Rafailidis and A. Nanopoulos, "Modeling users preference dynamics and side information in recommender systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 6, pp. 782–792, Jun. 2016.

[25] J. Xu, T. Xing, and M. van der Schaar, "Personalized course sequence recommendations," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5340–5352, Oct. 2016.

[26] W. Jian, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl.*, vol. 69, pp. 29–39, Mar. 2017,

[27] X.-L. Xu and G.-L. Xu, "Improved collaborative filtering recommendation based on classification and user trust," *J. Electron. Sci. Technol.*, vol. 14, no. 1, pp. 25–31, 2016.

[28] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowl.-Based Syst.*, vol. 26, pp. 225–238, Feb. 2012.

[29] Y. Du, X. Du, and L. Huang, "Improve the collaborative filtering recommender system performance by trust network construction," *Chin. J. Electron.*, vol. 25, no. 3, pp. 418–423, May 2016.

YING GAO received the M.Ed. degree from Liaoning Normal University, Dalian, China, in 2010. She is currently pursuing the Ph.D. degree with Nankai University. She is currently an Associate Professor with the Inner Mongolia University for the Nationalities. Her research interests include education economics and management, artificial intelligence algorithm, national education, and organizational structure of universities.

LINGXI RAN was born in Taian, Shandong, China, in 1983. He received the Engineering degree in information security from the Harbin Institute of Technology, Harbin, Heilongjiang, China, in 2008. He is currently pursuing the Doctor of Engineering degree in mechanical engineering with Shandong University, Shandong, China. His research interests include big data, artificial intelligence algorithm, and innovative design methodology.