FARM: A Fairness-Aware Recommendation Method for High Visibility and Low Visibility Mobile APPs

QILIANG ZHU, QIBO SUN, ZENGXIANG LI, AND SHANGGUANG WANG, (Senior Member, IEEE)

1College of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China
2State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
3Institute of High Performance Computing, Agency for Science Technology and Research (A*STAR), Singapore 138632

Corresponding author: Qiliang Zhu (zhuqiliang@ncwu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61571066.

ABSTRACT The number of mobile applications (APPs) has increased dramatically with the development of mobile Internet. It becomes challenging for users to identify these APPs they are really interested in. Existing mobile APP recommendation methods focus on learning users’ preference and recommending high visibility APPs. However, some low visibility APPs may satisfy users and even surprise them. If those low visibility APPs have the opportunity to show to the user, they will not only improve the user’s satisfaction, but also provide a fair competitive market for APP providers. Furthermore, it will improve the vitality of the APP market. To this end, we present a fairness-aware APP recommendation method named FARM. The principal study of this method emphasizes on the fairness issue during the recommendation process. In this method, APP candidates are divided into high visibility and low visibility APPs, and implement recommendation algorithm respectively. For low visibility APPs, we set a fairness factor for everyone, and use the user’s latest feedback to make a dynamic adjustment. Based on the fairness factor, the recommendation is implemented by roulette-wheel. For high visibility APPs, we employ the fuzzy analytic hierarchy process to implement the recommendation. The evaluation results show that FARM outperforms baselines in terms of recommendation fairness.

INDEX TERMS APP recommendation, fairness, roulette-wheel, fuzzy analytic hierarchy process.

I. INTRODUCTION

In recent years, the rapid growth of intelligence terminal has inspired the development of application (APP) market. Massive mobile APPs appear on the mobile Internet. In an APP market scenario, many APP providers may offer diverse APPs that match similar functional descriptions, leading to the problem of information overload. While enjoying the convenient services brought by mobile APP, users are also troubled by the huge amount of information, which makes it difficult for them to find the APP they really need. Therefore, how to assist users to find appropriate APPs becomes a severe challenge, and the ability to recommend relevant APPs to users is very urgent for APP market.

As one of the most effective ways to solve information overload, recommender system can assist users to find suitable items quickly [1]. A large number of recommendation approaches have been put forward for QoS prediction based on historical evaluation information. Collaborative filtering algorithm as one of the most common approaches for QoS prediction, has achieved great results [2-5]. However, some low visibility APPs which lack of sufficient historical evaluation information, can not get a fair chance in these classical recommendation approaches. In reality, these low visibility APPs may also satisfy requirements of users. However, these APPs do not have an opportunity to show to users, because their owners lack the ability or financial resources to advertise. For example, FnPlayer(1.0) is a low visibility video player in APP market. It is very difficult to find it except for special search. However, it has got a lot of praises after we recommended it to some volunteers. So, if these items
also have a chance to show to users, it will not only meet their needs, but also give APP provides a fairness competitive opportunity. Therefore, how to make these low visibility APPs to be recommended with an appropriate probability has been one of research focuses [7].

Numerous researchers deal with the recommendation problem of new APPs as a cold start problem [6,7]. Cold start technology employs similarity measurement and other approaches to help newborn APPs get QoS prediction. However, it can not provide a relatively fair opportunity for newborn APPs and low frequency APPs. To this end, we present a fairness-aware APP recommendation method named FARM. The focus of our study is providing fair recommended opportunities for low visibility APPs. In order to guarantee fairness during APP recommendation process, the APP candidates are divided into high visibility APPs and low visibility APPs, and recommendations are implemented respectively. For low visibility APPs, we set a fairness factor for every one, and implement APP recommendation based on roulette-wheel. For high visibility APPs, the fuzzy analytic hierarchy process is used for ranking prediction, and we make recommendation based on the predicted ranking. The major contribution of this work are threefold.

• We discuss the fairness problem in APP recommendation, and propose different recommendation strategies for high visibility and low visibility APPs.
• We design a fairness-aware APP recommendation method, and employ roulette wheel to consider fairness of low visibility APP during the recommendation process.
• We evaluate the proposed method by employing a real-world dataset. Experimental results demonstrate that FARM is superior to other state-of-the-art baselines in terms of fairness.

The remainder of this paper is structured as follows. Section II discuss related work of APP recommendation and introduce relevant concepts. Section III describes the proposed FARM in detail. In Section IV, we validate the performance of FARM through a series of experiments. Section V discuss the actual effect of the proposed method. Finally, conclusions and future work are presented in Section VI.

II. RELATED WORK

In this Section, we overview the research work related to the proposed method, such as recommendation algorithms, APP recommendation and fairness issues.

Recommendation technology has attracted great attention from academia, and various recommendation methods have been proposed, such as collaborative filtering recommendation methods [3, 4], content-based recommendation methods [8], and hybrid recommendation methods [9]. As a key topic of recommendation methods, QoS prediction has been widely investigated in academic circles [2], [3], [10]. Sun et al. [10] presented a high precision QoS prediction method, which can better reduce the negative impact of volatile data. Zheng et al. [3] combined the user-based Pearson correlation coefficient and the item-based Pearson correlation coefficient to achieve better QoS prediction accuracy. In [11], considering all the QoS dimensions, tensor decomposition method is used to predict the QoS value in high dimensional space. They modeled multidimensional QoS data as a tensor and calculated its composition matrix by decomposing the tensor. Chen et al. [12] utilized the characteristics of QoS to divide users into multiple regions. On the basis of regional characteristics, they proposed an improved nearest neighbor approach for QoS prediction. Although QoS prediction is one of the most effective methods of recommendation, these methods are mostly for Web Services.

Many approaches [13-17] have been put forward for mobile APP recommendation. For instance, Liu et al. [13] employed latent factorization algorithm to make a tradeoff between user privacy preference and APP functionality. This model took the privacy-sensitive privileges and rating matrix to learn the privacy preference and interest of each user, as well as the functionality of each APP, so as to predict user’s rating of APPs. AppJoy [14] utilized collaborative filtering algorithm to provide personalized recommendations for users by analyzing whether users actually use the APPs they install. Kim et al. [15] believed that the more users share the attributes of APP, the more similar their preferences for APP. They find similar users by measuring the similarity between the target user and his social members, thus recommending the appropriate APP for the target user. In [16], the authors presented a general framework of APP recommendation system using system-level collaboration strategy and multi-objective optimization method. They used a new particle swarm optimization algorithm to solve the optimization problem caused by the collaborative recommendation. Although all these algorithms have achieved good results, they are based on a large number of user records and other information. It is difficult to predict APP rating or achieve better recommendation results for APPs lacking evaluation information.

Data sparsity is another key issue in recommendation field, some works focus on this problem in APP recommendation. Liu et al. [6] utilized the nascent description of APPs to address the cold start problem caused by sparse data. By combining Twitter followers’features with APP’s descriptive features, the preference of target users can be accurately predicted. In [7], the authors presented an actual-attempting model to implement the decision making process of APP recommendation. They introduced concepts of actual satisfactory value and tempting value to each APP, and the recommendation results need to consider both user’s preference and app contest. This method can help new APPs get recommended opportunities, but it doesn’t guarantee the fairness of these opportunities. Meanwhile, the APP that really meets user preference can also be replaced.

The issue of fairness has been studied in different fields [22, 23], and achieved some results. In recent years, the issue of fairness in the field of service recommendation has begun to attract the attention of some scholars [24, 25],
III. OUR PROPOSED METHOD

This section describes the main idea of the proposed method. As described in Figure 1, APP candidates are divided into two groups, i.e., high visibility APPs and low visibility APPs, and then perform corresponding recommendation algorithms. For low visibility APPs, we extract APP features and design a fairness factor for each one, then employ roulette-wheel for recommendation. If an APP has been installed and used by a user, the user’s feedback will be employed to dynamically adjust the fairness factor. For high visibility APPs, we extract APP features and employ fuzzy analytic hierarchy process to carry on the APP ranking prediction.

The final recommended APPs may include both low visibility APPs and high visibility APPs. The number of low visibility APPs is \( n_l = \gamma \cdot n_s \) (0 \( \leq \gamma \leq 1 \)) and the number of high visibility APPs is \( n_m = (1 - \gamma) \cdot n_s \), where \( n_s \) is the length of recommendation list.

A. APP CLASSIFICATION AND DESCRIPTION

A number of candidate APPs can be obtained from the mobile APP market. Among these candidates, there are high visibility APPs and low visibility APPs. Firstly, we give a detailed description two kinds of APPs as follows.

1) HIGH VISIBILITY APP

An APP is high visibility APP, if it has been used by users, and the number of user evaluations (\( N_i \)) meets the conditions: \( N_i \geq T \), where \( T \) is the threshold of the number of user evaluations. Based on the properties of high visibility APP, it can be described as follows,

\[
S_m = \{id, pop, cre, num, \bar{r}, sta\},
\]

where \( S_m \) represents a high visibility APP, which can be described by six fields. \( id \) is the ID number of an APP, \( pop \) is the popularity of an APP, \( cre \) denotes the APP reputation, which represents the security of the APP and whether it has been embedded advertising, \( \bar{r} \) is the average rating given by users, \( num \) is the number of user records, \( sta \) denotes the stability of the APP, which denotes the rating difference from different users.

2) LOW VISIBILITY APP

An APP is low visibility APP, if the number of user evaluations meets the conditions: \( N_i < T \), meanings of \( N_i \) and \( T \) are the same as previously. Based on properties of low visibility APP, we describe it as follows,

\[
S_l = \{id, t, cre, num, \bar{r}\},
\]

where \( S_l \) represent an low visibility APP, which can be described by five attributes. \( t \) denotes the release time of the APP. The meaning of other parameters (\( cre, \bar{r}, num \)) are consistent with the previous. Specially, when \( num = 0 \), the APP is a new one, we can describe it as:

\[
S_n = \{id, t, cre\}.
\]

Obviously, \( T \) is a key parameter in the above classification, which has a great influence on the results of recommendation. In general, the value of \( T \) needs to be adjusted according to the data set.

B. LOW VISIBILITY APP RECOMMENDATION

For low visibility APP, there is nothing or just a little information can be utilized. This is also an important reason why the low visibility APP is often neglected during the APP recommendation process. Although there are a number of cold start methods to solve this problem, the recommendation opportunity is still not fair to the low visibility APP. We employ fairness factor to ensure that low visibility APPs can have a relatively fair opportunity during the recommendation process. In order to give all low visibility APPs a
reasonable probability, we first initialize fairness factor for all low visibility APPs. Then, we adjust it according to user’s latest feedback. Finally, we employ roulette-wheel algorithm for APP recommendation based on fairness factor.

1) FAIRNESS FACTOR INITIALIZATION
In this phase, the properties of low visibility APP are considered to design fairness factor. On the basis of equal opportunities, the recommendation opportunity of each APP should be attenuated over time. And the reputation and evaluation should be considered. Therefore, the fairness factor for every APP is defined as follows,

\[ v_f = \beta \left( \frac{1}{n_{im}} e^{-\lambda t} + \alpha \cdot cre \right) + (1 - \beta) \frac{(n_u - n_{min})}{n_{max} - n_{min}}, \]

where \( v_f \) denotes the fairness value of an low visibility APP, \( n_{im} \) is the number of low visibility APP candidates. \( e^{-\lambda t} \) denotes an attenuation coefficient for the time. \( \lambda \) is an attenuation coefficient. \( t \) is the release time of APP. \( cre \) is the reputation value of an APP, and \( \alpha \) is its weight. \( n_u \) represents the total number of evaluation records. \( n_{max} \) and \( n_{min} \) represent the maximum and minimum number of rating records in all APPs. \( \bar{r} \) denotes the average of all user ratings. \( \beta \) is a weight coefficient. Specifically, if an APP has just been released, it will have no user records. In this case, we set the value of \( \beta \) to 1.

\textbf{Algorithm 1} Fairness Factor Dynamic Adjustment

\begin{algorithm}
\begin{algorithmic}
\State \textbf{Data:} The set of low visibility candidates, \textit{incand\_set};
\State \quad The set of fairness factor value for each APP, \textit{vf};
\State \quad The recommended number of each APP, \textit{t_i};
\State \quad The set of user evaluation, \textit{r}.
\State \textbf{Result:} The adjusted set of candidates, \textit{incand\_set};
\State \quad The adjusted set of fairness factor value for each APP, \textit{vf};
\State \State \State n \leftarrow \textit{incand\_set.size}()
\State \State \State \textbf{for} each \textit{i} \textbf{=} 1 \textbf{to} \textit{n} \textbf{do}
\State \State \State \State \textbf{if} \textit{r}^i(\textit{i}) \neq \phi \textbf{then}
\State \State \State \State \quad \textit{vf}(\textit{i}) \leftarrow \textit{vf}(\textit{i}) \cdot e^{(\textit{r}^i(\textit{i}) - \textit{r}_{\text{mid}})/\textit{r}_{\text{max}}}
\State \State \State \State \textbf{end}
\State \State \State \State \textbf{if} \textit{r}^i(\textit{i}) < \textit{r}_{\text{mid}} \& \& \textit{r}^{i-1}(\textit{i}) < \textit{r}_{\text{mid}} \textbf{then}
\State \State \State \State \quad \textit{vf}(\textit{i}) \leftarrow 0
\State \State \State \State \textbf{end}
\State \State \State \State \textbf{if} \textit{t_i}(\textit{i}) \geq \textit{T} \textbf{then}
\State \State \State \State \quad \textit{incand\_set}(\textit{i}) \leftarrow \phi
\State \State \State \State \quad \textit{vf}(\textit{i}) \leftarrow \phi
\State \State \State \State \quad \textit{t_i}(\textit{i}) \leftarrow \phi
\State \State \State \State \quad \textit{r}(\textit{i}) \leftarrow \phi
\State \State \State \State \textbf{end}
\State \State \State \State \textbf{return} \textit{incand\_set};\textit{vf}
\end{algorithmic}
\end{algorithm}

Through the previous steps, we can get an initial value of the fairness factor for each low visibility APP. In fact, we don’t know much information about low visibility APPs, however, user’s latest feedback can give us a guidance. To improve the accuracy of fairness factor, users’ latest feedback is used as the basis for dynamic adjustment of fairness factor.

2) FAIRNESS FACTOR DYNAMIC ADJUSTMENT
If an APP is recommended and used by a user, it may satisfy user, or let user disappointed. The user will give a feedback, which is helpful to adjust the fairness factor value. Algorithm 1 is designed to achieve fairness factor dynamic adjustment. In this algorithm, we employ Eq.5 to adjust the fairness factor value.

\[ v_f^{t+1} = v_f^t \cdot e^{(r^t - r_{\text{mid}})/r_{\text{max}}}, \]

where \( v_f^{t+1} \) is the value of fairness factor at the \( t + 1 \) recommended time, and \( v_f^t \) is value of fairness factor at the \( t \) recommended time. \( r^t \) represent the feedback given by a user, \( r_{\text{mid}} \) and \( r_{\text{max}} \) are the intermediate and maximum value of the feedback standards, respectively.

To ensure the popularity of recommendation results and prevent extreme cases, we set the fairness factor value of the two cases as follows,

\[ v_f^{t+1} = \begin{cases} 0 & \text{if } r^t < r_{\text{mid}} \text{ and } r^{t-1} < r_{\text{mid}} \\ 0 & \text{if } t_i > T_s \end{cases}, \]

where \( r^t < r_{\text{mid}} \text{ and } r^{t-1} < r_{\text{mid}} \) indicates that user’s feedback value is two successive times lower than the intermediate value. \( t_i \) represents the recommended number of an APP, and \( T_s \) denotes a threshold value.

Finally, the fairness factor is transformed into the form of probability. And the processing method as follows,

\[ f_i = \frac{v_i}{\sum_{k=1}^{N} v_k}, \]

where \( f_i \) is the probabilistic fairness factor of APP \( i \). \( v_i \) denotes the value of fairness factor of APP \( i \). \( N \) represents the number of low visibility candidate APPs.

3) APP RECOMMENDATION BASED ON ROULETTE-WHEEL
Based on the probabilistic fairness factor, we employ roulette-wheel algorithm for APP recommendation. Roulette-wheel selection is one of the most common algorithms being used in genetic algorithms and evolutionary algorithms [18]. It is a probabilistic algorithm, which selects an individual according to the probability associated with the individual. The roulette-wheel selection follows the rule that the better the individual fits, the better its chances of survival. During the selection process, there is no guarantee that the best one must be selected, but the probability is relatively large. And the weak individual may also be selected. Therefore, this algorithm provides a balance between the good individuals and weak individuals.

Algorithm 2 is the low visibility APP recommendation based on roulette-wheel. For each round of selection, we calculate the share of each APP by taking an APP’s fairness
value and calculate its percent of the total value. Next, a random value is generated for selecting an APP. If an APP is selected, it will be put into the recommendation list, and removed from the candidate set. Then, the algorithm proceeds to the next round until adequate low visibility APPs are recommended.

Algorithm 2 Low Visibility APP Recommendation

Data: The set of candidate APPs, imcand_set; The set of fairness factor value for each APP, \( v_f \); The number of recommended low visibility APPs, \( num \).

Result: The set of recommended APPs, sele_set.

1. for each \( i = 1 \) to \( num \) do 
   2. \( n \leftarrow \) imcand_set.size()
   3. MoveSelector \( \leftarrow \) rand(0, 1)
   4. Accumulator \( \leftarrow 0 \)
   5. for \( k = 1 \) to \( n \) do
      6. \( f(k) = \frac{v_f(k)}{\sum_{j=1}^{n} v_f(j)} \)
      7. Accumulator \( \leftarrow \) Accumulator + \( f(k) \)
      8. if Accumulator \( \geq \) MoveSelector then
         9. sele_set(\( i \)) \( \leftarrow \) imcand_set(\( k \))
         10. imcand_set(\( j \)) \( \leftarrow \phi \)
         11. \( v_f(j) \leftarrow \phi \)
      end
   end

13. return sele_set

C. HIGH VISIBILITY APP RECOMMENDATION

For high visibility APP, there is sufficient information to assess the quality of each one. Generally speaking, the quality of APP is determined by multiple attributes. Therefore, the high visibility APP recommendation can be solved by a multi-criteria decision-making algorithm. Analytical hierarchical process as one of the multi-criteria decision-making models can formulate the problem in hierarchical form. However, criteria are usually conflicting in traditional analytical hierarchical process, and the solution is largely dependent on the preference of experts. We employ fuzzy analytic hierarchy process to solve this problem. Fuzzy analytic hierarchy process transforms the matrix construction into fuzzy consistent matrix construction of analytical hierarchical process[19]. This method can make a more reasonable weight evaluation of criteria and decisions.

1) CONSTRUCTION OF HIERARCHY INDEX MODEL

Through the analysis and determination of evaluation indexes, the decision problem should be divided into three layers As described in Figure 2, recommendation model includes target layer, criterion layer and scheme layer.

The target layer is at the top level, representing the recommendation objectives. The criterion layer is in the middle, which consists of the key attributes that determine the quality of service. The scheme layer is at the bottom, representing all candidate APPs.

2) CONSTRUCTION OF PRECEDENCE RELATION MATRIX

Construct the priority matrix with reference to the relative importance of the indexes in each layer to the upper layer. The value of each element in priority can be expressed through the scale method shown in Table 1.

This method would solve the problem greatly which is difficult to quantify the criteria from 1 to 9. Through the result of inter-comparison between different indexes, the priority matrix of each layer can be described as follows,

\[
F = (f_{ij})_{n \times n} = \begin{pmatrix}
        f_{11} & \cdots & f_{1n} \\
        \vdots & \ddots & \vdots \\
        f_{n1} & \cdots & f_{nn}
\end{pmatrix}, \quad (8)
\]

where \( F \) is a priority matrix of a layer. \( n \) denotes the number of indexes in this layer, \( f_{ij} \) denotes the pairwise comparison value between index \( i \) and index \( j \).

According to this method, we do a pairwise comparison between the indexes of criterion layer described in Figure. 2, and get the priority matrix. And the priority matrix of scheme layer could be acquired by the same way.

3) APP RECOMMENDATION BASED ON WEIGHTS CALCULATION

Based on the priority matrix, the fuzzy consistent matrix \( R = (r_{ij})_{n \times n} \) can be constructed. First, the sum of each row in
matrix $F$ can be calculated by Eq. 9.

$$f_i = \sum_{k=1}^{n} f_{ik}, \quad (i = 1, 2, \ldots, n),$$

(9)

where $f_i$ is the sum of elements in row $i$. Then the fuzzy consistent matrix can be constructed through the transformation as follows,

$$r_{ij} = \frac{f_i - f_j}{2n} + 0.5.$$  

(10)

Next, calculate hierarchy sort by importance of various factors. The weight vector of $R$ is calculated by,

$$s_i = \left( \prod_{j=1}^{n} r_{ij} \right)^{\frac{1}{n}}$$

(11)

And the weight vector would be normalized as follows,

$$\tilde{s}_i = \frac{s_i}{\sum_{j=1}^{n} s_j}, (i = 1, 2, \ldots, n).$$

(12)

The weight vector $w$ can be obtained as follows,

$$w = (\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_n)^T.$$  

(13)

According to the above method, we can calculate the weight vector of criterion layer $w_0$. Then the weight vector of the bottom layer $w_p$ can be calculated by the same method, where $p$ indicates the element of the middle layer. The final weight of each candidate APP can be obtained as follows,

$$w_1 = (w_1^1, w_1^2, \cdots, w_1^5)^T \cdot w_0.$$  

(14)

Finally, we can get the recommendation result according to the weight ranking of APP candidates.

### IV. EXPERIMENTAL EVALUATIONS

We conduct a series of experiments to verify the performance of FARM, and compare it with other three methods by evaluation metrics of precision, recall and fairness.

#### A. EXPERIMENTAL SETUP

According to the experimental requirements, we crawled data sets from Anzhi Market\(^1\) by multiple time periods, which named AnzhiDataSet. The dataset contains 5 crawling data with different periods from 20160718 to 20161230, and each period consists of 3550 data records. As shown in Table 2, each APP data contains 16 attribute information. The ‘Categories’ of APP mainly includes reading, finance, photography, travel and video. The ‘popularity’ denotes the concern degree. ‘time’ represents the release time of the APP, based on which we can identify whether an APP is a new one. ‘A’, ‘B’, ‘C’, ‘D’ and ‘E’ represent different grades. According to these information, the stability of an APP can be obtained. And the reputation can be calculated by ‘safe’ and ‘ad’.

\(^1\)http://www.anzhi.com/

| No. | Attribute  | Description       |
|-----|------------|-------------------|
| 1   | ID         | APP’s ID          |
| 2   | name       | APP’s name        |
| 3   | safe       | whether it is safe|
| 4   | ad         | whether the ad is embedded |
| 5   | provider   | APP provider      |
| 6   | popularity | concern degree    |
| 7   | size       | size of the APP   |
| 8   | category   | category of the APP|
| 9   | time       | release time of the APP |
| 10  | A          | evaluation times of ‘A’ |
| 11  | B          | evaluation times of ‘B’ |
| 12  | C          | evaluation times of ‘C’ |
| 13  | D          | evaluation times of ‘D’ |
| 14  | E          | evaluation times of ‘E’ |
| 15  | score      | average score given by all users |
| 16  | records    | The number of evaluation records |

We compare FARM with other three existing methods:
- Multi-criteria recommendation (MCR). This approach uses AHP to rank the relative importance of candidate APPs, which is described in [21].
- Contest between satisfaction and temptation (CST). This approach considers both user’s preference and app contest, and recommendation results are achieved through contest between actual satisfactory value and tempting value [7].
- Popularity (POP). This approach is one of the most common APP recommendations, which generates recommendations based on the popularity of APPs. It is a default recommendation rule in the Anzhi market.

All experiments were implemented on an IBM server with Inter Xeon E5-2670 8-core 2.60GHz processor, 32G RAM, Windows 7, and all algorithms are implemented in matlab 8.3.

#### B. COMPARISON METRICS

Four metrics are employed to evaluate the performance of the proposed FARM, which are precision, recall, opportunity fairness and qualification fairness. In all experiments, we judge whether an APP is an excellent one only by its rating. New APP is not considered as an excellent one because they have no rating records.

1) Precision is the ratio of the true positives to all the positive results (including true positives and false positives).

The definition is as follows,

$$\text{precision} = \frac{tp}{tp + fp},$$  

(15)

where $tp$ denotes true positives, and $fp$ denotes false positives.

In this paper, precision refers to the excellent APPs among recommended APPs to all recommended APPs.

2) Recall is the ratio of positive cases that are correctly identified. The definition is as follows,

$$\text{Recall} = \frac{tp}{tp + fn},$$  

(16)
where \( fn \) represents false positive. In this context, recall refers to the excellent APPs in recommended APPs to all excellent APPs.

3) **Opportunity fairness** is a metric to measure the fairness of recommendation opportunity, which can be expressed by jain’s fairness index [20].

\[
J(a) = \frac{\left( \sum_{i=1}^{n} a_i \right)^2}{n \sum_{i=1}^{n} a_i^2}, \tag{17}
\]

where \( J(a) \) denotes Jain’s fairness of individual \( a \), \( n \) denotes the total number of candidates with the same function. \( a_i \) represents the number of recommendations of APP \( i \). \( a_i \) can be any positive integer.

4) **Qualification fairness** does not care about the number of recommended times, just considers whether an APP has ever been recommended. It can also be calculated by Eq.17. The value of \( a_i \) is only different from opportunity fairness. In this case, \( a_i = 1 \) or \( a_i = 0 \).

**C. COMPARISON RESULTS**

After many experiments to select the optimal parameter, we set \( T = 10 \). In these experiments, we observed precision and recall by conducting the recommendation 200 times. To observe the performance of fairness, we recommended different number of APPs from 800 APP candidates, and did the recommendation 200 times, changing the length of recommendation list from 10 to 100. We also investigated the effect of the number of recommendation times for fairness. We recommended 40 APPs in each recommendation process, and changed the number of recommendation times from 50 to 500 with a step value of 50.

1) **THE PERFORMANCE OF PRECISION AND RECALL**

Figure 3 shows the values of precision and recall with different methods for APP recommendation. Although POP is the easiest way for recommendation, it just considers the number of usage records, and don’t care about user rating, so it doesn’t get a good result. For CST, Some new APPs enter the recommendation list through competition, but these new APPs are not considered excellent APPs because they do not have user rating information. So the performance of CST is not ideal. Obviously, FARM is better than CST and POP.

Regrettably, FARM is not the best method in terms of recall and precision, slightly lower than MCS. The reason is due to the fact that we provide low visibility APP a fair opportunity during the recommendation process. As the same with CST, the new APP is not considered as an excellent one, but other low visibility APPs may be considered as excellent APPs. Therefore, the result is preferable to CST. In fact, these low visibility APPs including new APPs may be accepted by users. In other words, these low visibility APPs including new APPs may also be excellent APPs. Therefore, FARM should have a better performance in these two metrics. Furthermore, improving the recommendation fairness for low visibility APPs is our main focus.

2) **THE PERFORMANCE OF FAIRNESS**

To verify the performance of FARM in fairness, we change the length of the recommended list and the recommendation frequency to analyze the performance of four methods in opportunity fairness and qualification fairness. Figure 4, Figure 5 and Figure 6 describe the comparison results.
As described in Figure 4, with the increase of recommended APPs, the opportunity fairness for all methods are raised. And the advantage of FARM is more prominent with the increase of recommended number. That is because the greater recommended number the more opportunity low visibility APPs will be given. However, opportunity fairness for FARM is still poor when the recommended number increasing. The reason for this is that some of the excellent high visibility APPs are available during each recommendation process.

Figure 5 shows the qualification fairness with different number of recommended APPs. Obviously, FARM is significantly higher than other methods. This is because that FARM employs roulette wheel to help more low visibility APPs get recommended qualification, and some other poor high visibility APPs cannot obtain recommendation qualification, so the qualification fairness reaches the highest point (0.827) when the recommended quantity increases. CST method allows some new APPs to be recommended through a contest mechanism, but it does not guarantee the qualification for everyone. So the performance is obviously inferior to FARM. For other two methods, even if the recommended quantity increase in each round of recommendation, low visibility APPs are still neglected, only more high visibility APPs have been recommended, and a large number of low visibility APPs are still in an unfair position. Although their qualification fairness rise slightly, they still at a disadvantage.

As shown in Figure 6, with the increase of recommendation times, the qualification fairness of FARM is getting bigger and bigger. CST also has better performance, but it is obviously less than FARM. The results for other two methods are just a slight rise, and always at a lower value. For FARM, with the increase of recommendation times, more and more low visibility APPs may obtain recommendation qualification benefit from roulette-wheel. For CST, contest mechanism also helps some new APPs obtain recommended qualification, but this contest mechanism can not guarantee most APPs have a recommended qualification. For other algorithms, a large number of low visibility APPs are never eligible to be recommended, no matter how many times the recommendation process is executed.

3) STUDIES ON PARAMETER $\gamma$

$\gamma$ is an important parameter for the proposed FARM, which controls the proportion of low visibility APPs during the recommendation process. To analyze the effect of parameter $\gamma$, we give it different values and observe the effect on qualification fairness, precision, and recall.

As shown in Figure 7, the qualification fairness increases with the increasing parameter $\gamma$, and keeps stable when $\gamma$ is large enough. The reason for this is that the greater the parameter $\gamma$, the greater the recommendation opportunity for low visibility APPs. However, when $\gamma$ is large enough, most low visibility APPs are eligible for recommendation. So the qualification fairness reaches the highest point when the parameter $\gamma$ large enough. On the contrary, the precision and recall become smaller as $\gamma$ becomes larger. This phenomenon can be explained as follows, with increases of $\gamma$, the number of low visibility APPs increases, and there will be some new APPs in low visibility APPs, which not be considered as an excellent APP. Therefore, the value of precision and recall will descend with the parameter $\gamma$ going up. In fact, the value of precision and recall will be higher, because some of the new APPs are actually excellent APPs which not be considered as excellent APPs in this experiment.

V. DISCUSSION

According to the experimental results, we can observe that FARM has a significant effect on improving recommendation fairness. However, the performance of recommendation precision and recall are not ideal. This is because that the new APP in the recommended list is not regarded as an excellent APP. In practice, FARM is expected to have a better results in precision and recall, and thus, promote the healthy development of the APP market.

To further verify the effectiveness of FARM, we compared the results with real data. In the third experiment, we collected
two APPs’ data at five time nodes. As illustrated in Figure 8, two APPs (real1 and real2) have no record at the beginning. In other words, both the two APPs are new APPs. As time goes on, recommended records of real1 are more and more, while real2 has not changed. Therefore, we can infer that real1 has a good reputation, and the reputation of real2 is not good. FAR1 and FAR2 are the results of real1 and real2 executing 400 recommendations (Execute 100 times between every two time nodes) by FARM, respectively. As shown in the figure, employing the proposed FARM, both APPs get a recommended chance at the beginning. After that, real1 has gained more opportunities to be recommended, and real2 is no longer a chance to be recommended. This is because when adjusting the fairness factor dynamically, the fairness factor of real2 is set at 0 due to the lack of effective evaluation, and there is no chance to be recommended in the later recommendation process. Because of the high value of user feedback, the fairness factor of real1 increases continuously when the fairness factor is adjusted dynamically, so that more opportunities can be obtained. In summary, The proposed FARM method can help excellent APPs to show to user faster, and don’t give too many chances to the poor APPs.

VI. CONCLUSION

This paper presented a fairness-aware APP recommendation method for the purpose of taking into account the fairness between high visibility and low visibility APPs. In this method, APP candidates are divided into high visibility APPs and low visibility APPs, and implement recommendation algorithms, respectively. The main study emphasizes on the fairness of low visibility APP during recommendation process. For low visibility APPs, we continuously adjust fairness factor for everyone according to the latest users’ feedback, and do recommendation based on the roulette-wheel algorithm. For high visibility APPs, we employ the fuzzy analytic hierarchy process to implement recommendation. According to the experiments based on real datasets, the proposed method shows good performance in terms of fairness, which can discover excellent APPs among low visibility APPs rapidly, and improve user satisfaction.

Our proposed method is a pioneer attempt for fairness-aware APP recommendation. In the future, we will consider additional factors, such as user preference and mobile context, so as to improve the accuracy of recommendation and user satisfaction.

REFERENCES

[1] D. Das, L. Sahoo, and S. Datta, “A survey on recommendation system,” Int. J. Comput. Appl., vol. 160, no. 7, pp. 6–10, Feb. 2017.
[2] Y. Ma, S. Wang, F. Yang, and R. N. Chang, “Predicting QoS values via multi-dimensional QoS data for Web service recommendations,” in Proc. IEEE Int. Conf. Web Services, Jun. 2015, pp. 249–256.
[3] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “QoS-aware Web service recommendation by collaborative filtering,” IEEE Trans. Services Comput., vol. 4, no. 2, pp. 140–152, Apr. 2011.
[4] M. Tang, Y. Jiang, J. Liu, and X. Liu, “Location-aware collaborative filtering for QoS-based service recommendation,” in Proc. IEEE 19th Int. Conf. Web Services, Jun. 2012, pp. 202–209.
[5] Q. Yu, “QoS-aware service selection via collaborative QoS evaluation,” World Wide Web-Internet Web Inf. Syst., vol. 17, no. 1, pp. 33–57, Jan. 2014.
[6] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua, “Addressing cold-start in app recommendation: Latent user models constructed from Twitter followers,” in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. SIGIR, 2013, pp. 283–290.
[7] P. Yin, P. Luo, W.-C. Lee, and M. Wang, “App recommendation: A contest between satisfaction and temptation,” in Proc. 6th ACM Int. Conf. Web Search Data Mining WSDM, 2013, pp. 395–404.
[8] B. Malesza and M. Maleszka, “Influence of social communication on content-based recommendation,” in Proc. IEEE Int. Conf. Inf. Integr. Syst. Appl. (INISTA), Jul. 2017, pp. 143–148.
[9] S. Wei, X. Zheng, D. Chen, and C. Chen, “A hybrid approach for movie recommendation via tags and ratings,” Electron. Commerce Res. Appl., vol. 18, pp. 83–94, Jul. 2016.
[10] Q. Sun, L. Wang, S. Wang, Y. Ma, and C.-H. Hsu, “QoS prediction for Web service in mobile Internet environment,” New Rev. Hypermedia Multimedia, vol. 22, no. 3, pp. 207–222, Jul. 2016.
[11] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. N. Chang, “Multi-dimensional QoS prediction for service recommendations,” IEEE Trans. Services Comput., vol. 12, no. 1, pp. 47–57, Jan. 2019.
[12] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, “Personalized QoS-aware Web service recommendation and visualization,” IEEE Trans. Services Comput., vol. 6, no. 1, pp. 35–47, 1st Quart., 2013.
[13] B. Liu, D. Kong, L. Cen, N. Z. Gong, H. Jin, and H. Xiong, “Personalized mobile app recommendation: Reconciling app functionality and user privacy preference,” in Proc. 8th ACM Int. Conf. Web Search Data Mining WSDM, 2015, pp. 315–324.
[14] B. Yan and G. Chen, “AppJoy: Personalized mobile application discovery,” in Proc. 9th Int. Conf. Mobile Syst. Appl., Services MobiSys, 2011, pp. 113–126.
[15] J. Kim, S. Kang, Y. Lim, and H.-M. Kim, “Recommendation algorithm of the app store by using semantic relations between apps,” J. Supercomput., vol. 65, no. 1, pp. 16–26, Jul. 2013.
[16] X. Xia, X. Wang, J. Li, and X. Zhou, “Multi-objective mobile app recommendation: A system-level collaboration approach,” Comput. Electr. Eng., vol. 40, no. 1, pp. 203–215, Jan. 2014.
[17] K.-P. Lin, Y.-W. Chang, C.-Y. Shen, and M.-C. Lin, “Leveraging online word of mouth for personalized app recommendation,” IEEE Trans. Comput. Social Syst., vol. 5, no. 4, pp. 1061–1070, Dec. 2018.
[18] A. Limparski and D. Lipowska, “Roulette-wheel selection via stochastic acceptance,” Phys. A, Stat. Mech. Appl., vol. 391, no. 6, pp. 2193–2196, Mar. 2012.
[19] L. Wang, J. Chu, and J. Wu, “Selection of optimum maintenance strategies based on a fuzzy analytic hierarchy process,” Int. J. Prod. Econ., vol. 107, no. 1, pp. 151–163, May 2007.
[20] A. B. Sediq, R. H. Gohary, R. Schoenen, and H. Yanikomeroglu, “Optimal tradeoff between sum-rate efficiency and Jain’s fairness index in resource allocation,” IEEE Trans. Wireless Commun., vol. 12, no. 7, pp. 3496–3509, Jul. 2013.
[21] A. Pinandito, M. Ananta, K. Brata, and L. Fanani, “Alternatives weighting in analytic hierarchy process of mobile culinary recommendation system using fuzzy,” ARPN J. Eng. Appl. Sci., vol. 10, no. 19, pp. 8791–8798, 2015.

[22] H. He, H. Shan, A. Huang, L. X. Cai, and T. Q. S. Quek, “Proportional fairness-based resource allocation for LTE-U coexisting with Wi-Fi,” IEEE Access, vol. 5, pp. 4720–4731, 2017.

[23] S. Chakraborty and S. Nandi, “Distributed service level flow control and fairness in wireless mesh networks,” IEEE Trans. Mobile Comput., vol. 14, no. 11, pp. 2229–2243, Nov. 2015.

[24] E. L. Lee, J.-K. Lou, W.-M. Chen, Y.-C. Chen, S.-D. Lin, Y.-S. Chiang, and K.-T. Chen, “Fairness-aware loan recommendation for microfinance services,” in Proc. Int. Conf. Social Comput., 2014, pp. 1–4.

[25] K. Huang, Y. Liu, S. Nepal, Y. Fan, S. Chen, and W. Tan, “A novel equitable trustworthy mechanism for service recommendation in the evolving service ecosystem,” in Proc. Int. Conf. Service Oriented Comput., Springer, 2014, pp. 510–517.

QILIANG ZHU received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2018. He is currently a Lecturer with the College of Information Engineering, North China University of Water Resource and Electric Power. His research interests include service computing, edge computing, and recommender systems.

QIBO SUN received the Ph.D. degree in communication and electronic system from the Beijing University of Posts and Telecommunication, China, in 2002. He is currently an Associate Professor with the Beijing University of Posts and Telecommunications. His research interests include services computing, the Internet of Things, and network security. He is a member of the China Computer Federation.

ZENGIANG LI received the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2012. He is currently a Scientist with the Institute of High Performance Computing (IHPC), A*STAR. He is also a Principal Investigator of several projects involving multidisciplinary collaborations and real-world applications, such as urban computing, advanced manufacturing, and healthcare. His research interests include blockchain, secure multiparty computation, federate learning, the IoT systems, cloud computing, machine learning, large-scale data/graph processing, simulation and modeling, and parallel and distributed computing.

SHANGGUANG WANG (Senior Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications (B.U.T), in 2011. He is currently a Professor and the Deputy Director of the State Key Laboratory of Networking and Switching Technology, B.U.T. He has published more than 100 articles, and played a key role at many international conferences, such as general chair and PC chair. His research interests include service computing, cloud computing, and mobile edge computing. He is the Editor-in-Chief of the International Journal of Web Science.