Data Driven User Feature Construction and Requirement Generation in Mobile Applications

Yuanbang Li[1,*], Kai Wang[1], Xiaoyu Guo[1], Zhongyuan Hua[2], and Yong Xiao[2]
[1]Zhoukou Normal University, Zhoukou, Henan, China
[2]School of Computer Science and Artificial Intelligence,
Wuhan Textile University,Wuhan, Hubei,China
lybang@whu.edu.cn, 3101709158@qq.com, guoxiaoyu_sr@163.com,
1745224789@qq.com,634218476@qq.com

***Abstract*—Mobile applications have been widely used in people's daily life, and lots of data is generated in use of these mobile applications. These data including various context elements and contains rich user information. How to construct user features and discover user requirements from these data is vital for developer to improve the mobile application and providing smartly service for users. Therefore, a user feature construction and requirements generation method is proposed in this study. The main constructions of the study including three parts. Firstly, a time smoothing strategy is proposed to assist the generation of users' time-related features. Secondly, a fuzzy logic based method is proposed to generate user requirements. Thirdly, a tool is developed to automate and standardize the method. Finally, an experiment is conducted, and the results show the method is effective.**

Keywords-user feature; time smoothing; fuzzy logic; user requirement generation

I. INTRODUCTION

Owing to the maturity of mobile communication, mobile Internet is integrating and changing people's daily lives, works and learning environments. According to the 49th statistical report on the development of China's internet in 2022 by the China Internet Information Center [1], as of December 2021, the number of Chinese netizen reached 1032 million, of which 1028 million were mobile netizen, accounting for 99.7%. These netizens spend 28.5 hours online per week, up 2.3 hours from December 2020. With the widespread use of mobile phones with rich functions, a variety of mobile applications have sprung up like mushrooms [2]. These mobile applications have penetrated into all aspects of people's daily lives, and become a popular style for people to communicate, play, read, shop, socialize, schedule and work [2-5].

Mobile applications are facing fierce market competition. On the one hand, the competition of similar applications is fierce, and the proliferation of homogeneous applications has brought great challenges to the sustainable development of mobile applications [6]. On the other hand, Maslow's hierarchical requirements theory shows that people will and will necessarily pursue higher-level requirements when the lower-level requirements are satisfied. The same is true for mobile applications, when new functions of mobile applications (such as planning trips, shopping recommendations, online car-hailing, etc.) that meet people's requirements appear, people's attitudes towards these new functions will soon change from euphoria to accustomed, which will lead to new expectations. If a mobile application fails to capture user requirements in a timely manner, it will soon be replaced by other applications. Therefore, only by focusing on user experience and user requirements can the mobile applications continuously meet user expectations and stand out in the fierce competition.

It is difficult to extract personalized requirements of users in mobile applications; the reasons are shown as follows:

- User transparency

In the traditional software development process, developers can communicate with users directly to acquire their requirements. However, mobile applications do not have a specific user group, they are published in mobile application stores after the development, and can be downloaded freely by public users. Developers cannot communicate with users directly, and lack understanding of users, which increases the difficulty of obtaining user requirements.

- Low user stickiness

User feedback is an import way for software improvement, however, user's stickiness is low and they are negative to give comments. When user is interested in the mobile application, they will use it, when they find the application is unappealing or even disappointing, they will uninstall the application and switch to other similar applications which are more attractive.

- Complex environment

The operating environment of traditional software is stable, however, the operating environment of mobile application is complex and dynamically, users have different requirements in different environment such as time, location or weather, and it is difficult for users to accurately describe what they want and need to do in situations that they cannot directly contact in life [7-8].

Many requirements acquisition methods have been proposed to solve these challenges. Such as Niemen and Pasquale et al. [9-10] using observation method to obtain user requirements, they first obtained first-hand information by observing the behavior, attitude, environment and trajectory of users, and then combined with interviews and group discussions with domain experts to analyze and reason the requirements of users. Benabbou and Gregoriades acquire user requirements in complex environment through the formal definition or simulation of scenario [11-12]. Niemann M. and Li X [13-14] acquire user requirements based on the analysis of user comments in mobile applications.

2693-9371/22/$31.00 ©2022 IEEE
DOI 10.1109/QRS-C57518.2022.00096

The researches mentioned above using observation method, scenario method and comment analysis method to acquire user requirements, however, there is a lack of research on behavior data analysis to acquire user requirements. In this study, a user requirements acquisition method based on user behavior data is proposed, and the contributions are described as follows:

(1) a user feature construction method is proposed to build digital user feature matrix based on user behavior data.

(2) A temporal smoothing method is proposed to improve the accuracy of user time-related feature.

(3) A fuzzy logic-based approach is proposed to generate user requirements.

(4) A tool is developed to realize the methods.

The remainder sections in the manuscript is outlined as follows: the related works are summarized in section II. The methods proposed in this study are described in section III. The experiments are conducted and discussed section IV. The conclusion and future research directions are discussed in Section V.

II. Related Works

Current requirements acquisition methods in mobile applications including observation method, scenario-based method and comment analysis method.

(1) Observation method

Observation method is one of the basic methods of requirement extraction, and it has become a common requirement acquisition method in mobile applications because it can obtain first-hand information by observing the behavior, attitude and environment of users. Although the observation method can accurately capture the user's environment and behavioral trajectory, the requirements behind the user's behavior can only be obtained by inference. Therefore, in the requirement extraction of mobile applications, the observation method is often used in conjunction with interview method, questionnaire method, and group discussion method[9-10], that is, relevant users are first organized to participate in interviews or discussions to obtain the expected behavior of users, then compare these behavior with the actual behavior obtained by observation method, and user requirements are identified through the deviation analysis between the two [15]. For the situation where only the user's behavior can be observed but their intention cannot be identified, a controlled small-scale group can be selected to implement the above process to obtain the corresponding relationship between the user's situation, behavior and intention, and then use it as training data, and machine learning methods such as conditional random field [16] are introduced to infer user intent behind behaviors, thereby capturing the requirements of users.

(2) Scenario-based method

It is difficult for users to describe the requirements clearly before using the mobile application because of the dynamic and complexity of its running environment. The scenario-based method can help users to understand the system and express their requirements. Therefore, it has been widely used in requirement extraction of mobile applications. The scenario-based method refers to a method that takes the scenario as the core and extracts the requirements by defining the scenario, analyzing the scenario and simulating the scenario.

- Defining the scenario

Since the context elements involved in mobile applications are complex, the method needs to focus on describing the context elements related to user requirements, abstract these elements to generate a scenario to guide the acquisition of user requirements. UML is feature-rich and widely used, therefore, extending UML to describe scenarios in mobile applications has become a popular choice of many research [11,17-18]. [17] extended the UML sequence diagram and introduced two types of scenario participants, one is a functional participant, the other is a situational participant, and the user requirements of mobile applications are extracted by analyzing the interaction process of the participants. In order to analyze the influence of contextual elements on the system state and state changes, the literature [11] introduced an event-triggered analysis process in the activity diagram, and extracts mobile application requirements by analyzing the contextual elements of a series of event-triggered conditions from the initial state to the end state of user’s activities.

- Analyzing the scenario

The method of analyzing the scenario refers to proposing an easy-to-implement process or developing a tool to assist requirements acquisition with the scenario as the core in mobile applications [19-23]. Such as the literature [19] proposed a 5-step method for acquire requirements, including identifying relevant scenarios, defining available services, drafting scenario outlines, drawing scenario description diagrams, and generating scenarios and scenario requirements to achieve requirement extraction. In [20], a 3 step process is proposed, which are describing the scenario in summary, using the Carroll element to describe the scenario, and using the business process modeling notation to analyze the scenario. The Carroll element includes four categories: setting, participants, activities and events, these elements can be used to describe and analyze the scenarios to acquire requirements. In the literature [21-23], Scenario Presenter and Mobile Scenario Presenter are developed, these tools can obtain various environmental data automatically during the inspection process, so as to analyze the intention of mobile users to assist the requirements acquisition of mobile applications.

- Simulating the scenario

In the operating environment of mobile applications, there may be unsafe factors in some scenarios. In such scenarios, the security risks are great and user’s requirements are difficult to extract. It is very meaningful to use virtual reality technology to simulate the scenario to extract user’s requirements in this dangerous scenario. Such as simulating the influence probability of context elements to safe driving[12], the elements may including location of obstacles and whether the driver uses a mobile phone.

(3) Comment analysis method

Mobile users often express their problems and new ideas through comments [24], comments have become a major channel for them to express their experience and feedback, and are an important source of mobile application requirements. Some of these method using features extraction [25-26], topic modelling [27-28] to mining the features and topics from comments to generated user requirements. some of these methods using machine learning methods such as classification to analyze comments to identify user requirements [29-30].

Although these requirements acquisition methods have been proposed to acquire requirements in mobile applications. Few studies analyze behavioral data for user requirements acquisition. Therefore, an user requirements acquisition method based on user behavior data is proposed in this study.

III. METHODOLOGY

A. Overview

The framework of the method is shown in Figure 1. As shown in the figure, the method is on basis of user data, and it consist of two parts, the first part is user feature construction, which transform disordered data to user feature matrix. The second part is user requirement generation, which generate user requirements from user feature matrix using fuzzy logic.

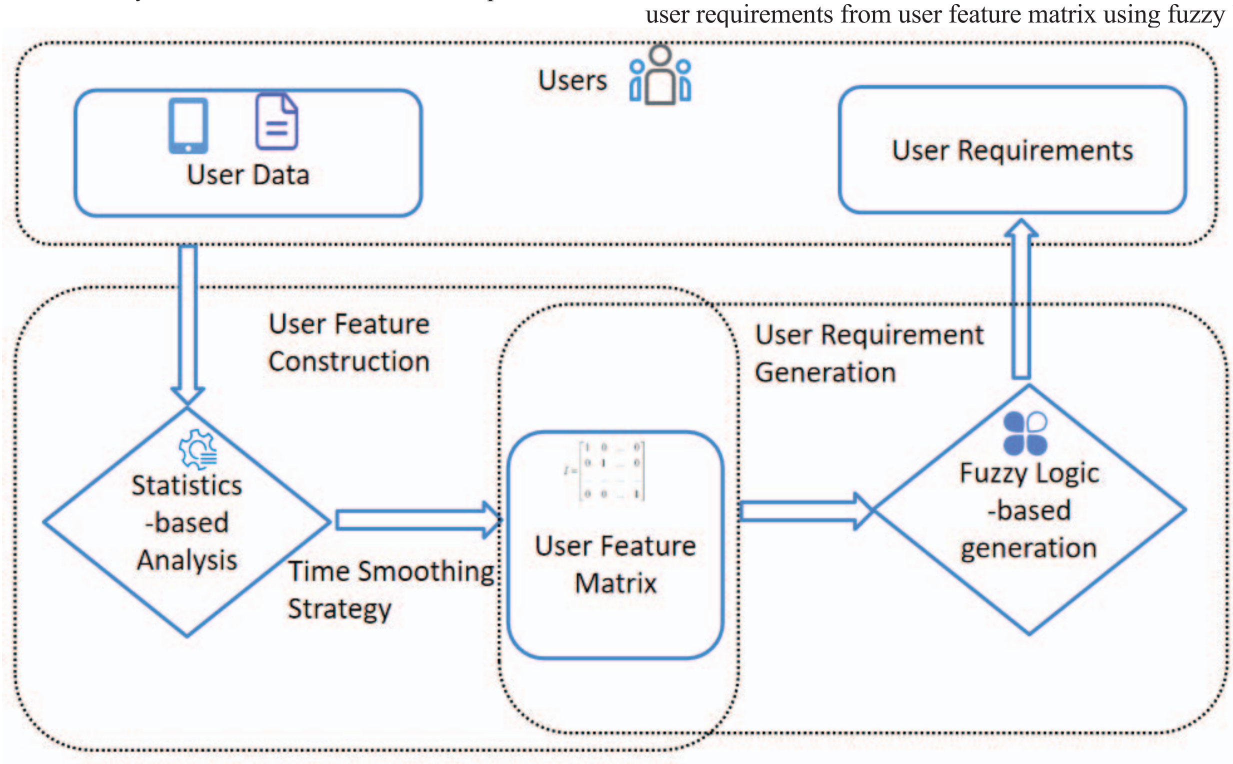

Figure 1. Framework of the method

B. User Feature Construction

Smartphone have powerful functions, and various sensors built in it can sense a variety of context elements, not only environmental elements such as time, location, humidity, air quality, etc., but also somatosensory elements such as heart rate, body temperature, etc. Different mobile application managers care about different topics. For example, mobile store managers care about when and where users buy what types of products, while doctors care about users' health status. Not all of the context elements have a significant impact on the topics, therefore, how to quantitatively analyze the influence of contextual elements on topics is the primary problem in the construction of user features.

In this study, an entropy based method is used to quantify the impact of context elements to topics.

The calculation of entropy is shown in formula 1, N denotes category numbers in the data, and P_i is the probability of belonging to category i.

$$Entropy(S) = -\sum_{i=1}^{n} p_i * log\, p_i \tag{1}$$

Entropy gain is effective to measure the effectiveness data partitioning. The entropy gain and entropy gain rate can be calculated in formulas 2 and 3.

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values}(A) \frac{S_v}{S} Entropy(S_v) \tag{2}$$

$$Gain\_ratio(S,A) = \frac{Gain(S,A)}{Entropy(S)} \tag{3}$$

In these formulas, a represents the attribute that divides the sample set S, Values represents the value set of the attribute A, v represents a value in the Values set, and S_v represents the sample set corresponding to the value v after the division.

The process to calculate the impact of context elements to topics is shown in algorithm 1.

```
Algorithm 1:  the impact calculation of context elements to topics
Input: DS(Data Set), UCET (User Context Element Set), TS (Topic Set)
Output: ES_S (Effect Set Selected)
initialize ES_S=NULL
 for each t in TS
   Computes Entropy (t) based on DS using formula (1)
   for each uce in UCES
     Computes Gain (t, uce) using formula (2)
     Computes Gain_ratio (t, uce) using formula (3) if
     if(Gain_ratio (t ,uce)) >δ
        ES_S.add (<V^j ,UC^i>)
     endif
   endfor
endfor
return ES_S
```

After the impact analysis of the context elements to the topics based on the entropy gain, the user feature matrix is construct for each effective combination. The number of the matrix row is the number of values of the context element, and the number of matrix columns is the number of values of the topic. The construction process is detailed in the previous work, and will not be repeat here [31].

Time is an important contextual element in user features. In order to clearly describe the time-related features of a user in one day, time is segmented in hours in this study. In existing research, time is usually handled in a discrete manner, that is to say using the hours of check-in time to construct user features [32-33]. For example, when the user's check-in time is 08:00:00, this check-in is considered to represent user's feature at 8 o' clock. When the user's check-in time is 08:59:50, it also be considered to represent user's feature at 8 o' clock. However, time is continuous, it may be inaccurate to calculated in discrete, a check-in activity at 08:59:50 has a greater impact on user's feature at 9 o'clock than 8 o'clock. Therefore, on the basis of discrete time segment division, a temporal smoothing method is used to assist the construction of user's time related features in this study.

The smoothing method in this study smooth a check-in to adjacent hours, that is, the check-in no longer represents for a certain hour, but represents the contribution to adjacent hours. Assuming that the time of a check-in is h:m:s, where h represents hours, m represents minutes, and s represents seconds, the impact of this check-in on user features is shown in Equation 4. When calculating count(h), the value of m Represents the difference between the check-in time and h point, and the proportion of the difference is m/60, therefore, the contribution of check-ins to h point is 1-m/60. When calculating count(h+1), the time difference between h and h+1 is 1 hour, corresponding to 60*1 minutes, then the difference between m minutes and h+1 point is 60*1-m minutes, and the proportion of the difference is (60*1-m)/60, so the impact of this check-in on h+1 hour is 1- (60*1-m)/60, and using h+1 to calculate the remainder of 24 is to solve the situation that h+1 is greater than or equal to 24.

$$\begin{cases} Count(h) = Count(h) + (1 - \frac{\mathrm{m}}{60}) \\ Count\big((h+1)\%24\big) = Count((h+1)\%24 + (1 - \frac{60*1-m}{60}) \end{cases} \tag{4}$$

For example, if the user checks in at time 12:15:00, after using the time smoothing method, it is calculated according to formula 4. The impact of the check-in on adjacent time segments is: Count (12) =Count (12) +0.75, Count (13) = Count (13) +0.25. Seconds are also included in the check-in time, but the impact of seconds is 1/60 to minutes. Experiment shows that seconds have a limited impact on user time feature, so it is not considered.

C. User requirements generation

User features in matrix can be directly calculated and analyzed, but it is not conducive to intuitive understanding of user requirements, therefore cannot be communicate directly with end users. So it is necessary to describe user requirements in natural language. In this section, based the calculation of user features, a fuzzy logic based method is used to realize the natural language description of user's requirements, which is described as follows.

Definition 1: requirements set RS

Requirement set, labeled RS, is a quadruple defined as:

RS: = {U, UCES, TS, FS}

Among them, U represents the set of users, UCES represents the set of user context elements, TS represents the set of different topics. And FS represents the set of fuzzy descriptions to describe weather the requirement is established by fuzzy logic calculation, which is specified in definition 2.

Definition 2: Fuzzy description set FDS

The fuzzy description set FDS is a fuzzy description of whether the requirements in R are valid, and is specifically defined as:

FS: = <FD:MD>

Among them, FD means fuzzy description, which is described by language set {low, medium, high} in this study. MD represents the membership degree.

When the context element is uce (uce∈UCES) and the topic is t (t∈TS), the maximum value of different user check-ins is defined as max, the minimum value is defined as min, and the average value is defined as avg, then the calculation of membership degree of low, medium and high is shown in formulas 5, 6 and 7.

$$M\_d(\text{low}) = \frac{\frac{avg}{2} - u\_c}{\frac{avg}{2} - min},\ u\_c < \frac{avg}{2} \tag{5}$$

$$M\_d(med) = \frac{u\_c - \frac{avg}{2}}{avg + \frac{max - avg}{2} - \frac{avg}{2}}, \frac{\text{avg}}{2} \le u\_c < avg + \frac{max - avg}{2} \tag{6}$$

$$M\_d(high) = \frac{u\_count - (avg + \frac{max - avg}{2})}{max - (avg + \frac{max - avg}{2})}, u\_c \geq avg + \frac{max - avg}{2} \quad (7)$$

In this paper, it is assumed that when the number of user's activity is less than half of the average, the probability of users participating in this activity is low. When the number of user activity is higher than half of the average and lower than the midpoint between the maximum and the average, the probability of the user's participating in this activity is medium. When the number of user activities is higher than the midpoint between the maximum value and the average value, the probability of the user participating in this activity is high.

IV. EXPERIMENTS

Experiments are designed to verify the effectiveness of the method. The content in this section includes: data sets of the experiment, results and analysis.

These experiments are conducted on a personal computer, the CPU of the computer is AMD Ryzen 5 4500U with Radeon Graphics, 2.38 GHz, the memory is the computer is with 16 GB, and the operating system of the computer is Win 10 with 64-bit. These experiments are implemented using Java language, Eclipse integrated development environment and MySQL database.

A. Date sets of the Experiment

The data set used in [34-35] is adopted in this paper, as it is widely used and public available. The data set in [34] is generated based on the check-in data from New York, which is denoted as Ds-1. We also randomly selecting 5000 and 8000 users from the data set in [35] to shown that the method is not affect by different data sets, which is denoted as Ds-2 and Ds-3. Overview of the data sets are shown in Table 1, and the illustration of the data are shown in Table 2.

Table 1 Overview of the data sets

	User number	POI number	Check-in times
Ds-1	1083	38333	227428
Ds-1	5000	359036	1472935
Ds-2	8000	509440	2253379

Table 2 Illustration of the data

U	P	PC	PCN	LO	LA	Week	Year	Month	Day	Time
9	43*	4c*	stadium	40.78*	-73.9*	Sun	2012	May	06	11:21:17
88	4b*	4f*	movie theater	40.75*	-73.9*	Fri	2012	Aug	17	08:13:18
…	…	…	…	…	…	…	…	…	…	…
126	42*	4a*	bar	40.76*	-73.9*	Wed	2012	Dec	21	17:21:42

Notes that in the table, U denotes user number, P denotes point of interest, PC denotes the category of point of interest, PCN denotes category name of the point of interest, LO denotes longitude and LA denotes latitude. The * is a shorthand of a long string.

POI category is included but POI root category is not in the original data set. To provide a clear understanding of user features in abstract, the POI root category is add to the data according to the relationship between the category and the root category in the category hierarchy tree on the Foursquare website. There are nine root categories in the data, which are arts and entertainment, college and university, food, outdoors and recreation, professional and other places, residence, shops and services, travel and transport, and events.

B. Results and analysis

Based on the analysis of the data, the context elements including time and location, and the topic of the manager including POI category and POI root category. Time and POI category can be extracted from the data directly, the POI root category can be generated based on the relationship between the category and the root category. The context elements of distance should be calculated. Finding the location of user's home is the premise of calculating the distance. In this study, the progressive home recognition method [36] is used to discover user's home, then the distance from the POI to user's home is calculated. The distance is a discrete value, its meaning to users is unclear. Therefore, it is segmented into different levels [37-38], namely less than 1 km, from 1 to 10 km, from 10 to 30 km, and more than 30 km.

After the context elements and topics are determined, the impact of contextual elements on topics are quantitatively calculated, the results are shown in Table 3.

Table 3 entropy gain rate of context elements to topics

	Context elements	Poi root category	Poi category
Ds_1	Time	18.69%	21.70%
	Distance	20.71%	23.64%
Ds_2	Time	14.71%	15.33%
	Distance	16.32%	18.53%
Ds_3	Time	21.61%	23.54%
	Distance	21.76%	22.48.%

According to the value of the data, in hours, there are 24 possible values for time, 4 values for distance levels, 65 values for POI category and 9 values for root category. Therefore, user's time-category feature is a matrix with 24 rows and 65 columns, user's time-root category feature is a matrix with 24 rows and 9 columns, user's distance-category feature is a matrix with 4 rows and 65 columns and user's distance-root category feature is a matrix with 4 rows and 9 columns. An example of time-root category and time-category are shown in Figure 2 and Figure 3.

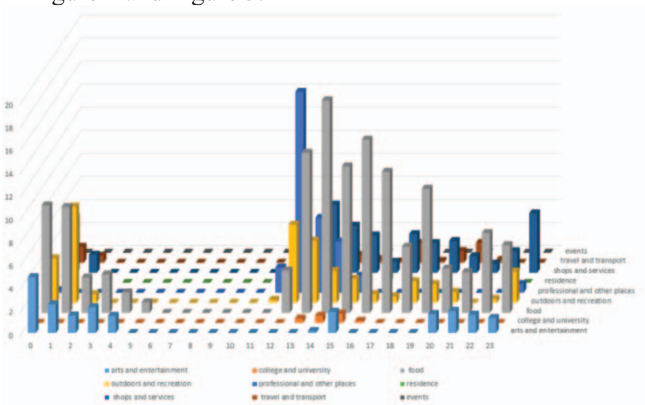

Figure 2. Example of user's time-root category feature

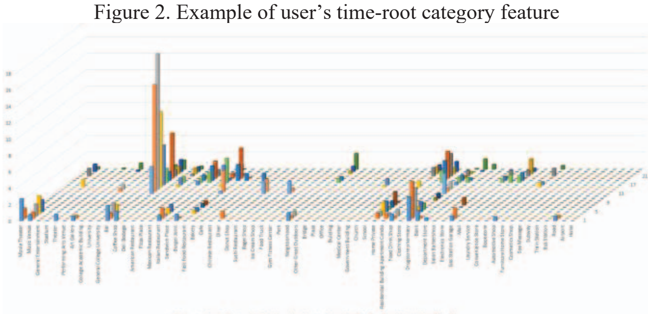

Figure 3. Example of user's time-category feature

A smoothing method is used in the construction of user's time related features. An experiment is conducted to verify the effectiveness of the method. In the experiments, for each data set, 80% is taken as the training set to construct user's features, and the remaining 20% is used as the test set. Because these features are established for each user separately, the division of the data set is also divided according to the data of each user, that is, the check-in data of each user is divided, and then take the union of all users' check-in data as the final data set.

In mobile applications, analysts are more concerned with providing the most suitable services to users, hence the main goal of the evaluation is to verify whether the recommended list of services is of interest to users. Therefore, we use the TOP K accuracy rate as the evaluation index, the calculation of the accuracy rate is shown in formula 8.

$$Accuracy@K = \frac{|\{u,l,t,a\}|a\in P_{u,l,t}(K),(u,l,t,a\in TS)|}{|TS|} \quad (8)$$

In the formula, {u,l,t,a} refers to an activity a of user u at time t at position l, $P_{u,l,t}(K)$ refers to the TOP K activity of the user at time t or position l inferred by the model, TS refers to the test set.

In the experiment, two user features are constructed, the first one is constructed using the hours in the time directly without time smoothing strategy. The second one is constructed using the smoothing strategy. And then, formula 8 is used to calculate the accuracy of the two user features. The experimental comparison results are shown in Figure 4 and Figure 5.

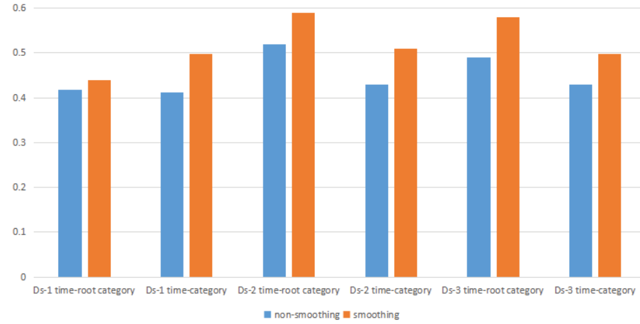

Figure 4. Top 1 accuracy of time-related feature

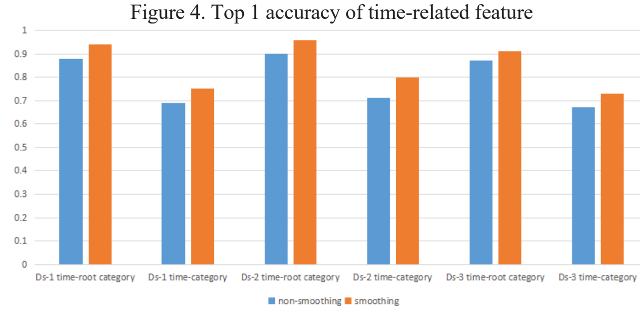

Figure 5. Top 2 accuracy of time-related feature

As shown in the figure, after adopting the smoothing strategy, the accuracy of user features in all three datasets are improved, which indicates that the smoothing strategy is effective to improve user's time-related features.

After the construction of user features, in basis of the user feature matrix, user requirement generation method based on fuzzy logic described in sub-section C of section III is used to generate user requirement documents. The user requirement documents include a total of 14083 users' requirements in 3 datasets.

In order to calculate the rate of fuzzy description using fuzzy logic, the maximum value and the average value of the user activities in the context element uce (uce∈UCES) and the topic t (t∈TS) need to be used. Taking Ds_1 as an example, its maximum and average values of time-category feature are shown in Figures 6 and 7. The maximum and average values in other datasets and other user features are similar and will not be shown repeatedly.

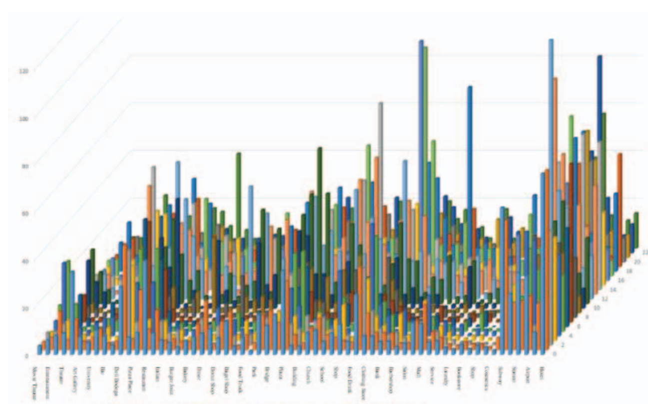

Figure 6. Max value of user's time-category feature

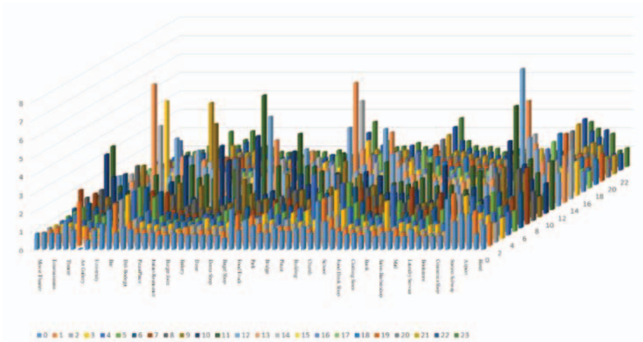

Figure 7. Average value of user's time-category feature

After the maximum and average value of user' feature are calculated, user requirement is generated. Because the values of context elements time and distance are 24 and 4 respectively, and the values of topic category and root category are 65 and 9, therefore the maximum number of requirement items generated by the combination of these two is 2072. Because too many requirement items will overwhelm the main requirements of users, and requirements with too low probability are meaningless to users, the user requirements are filtered in this paper, the items with probability lower than half of the average are deleted. An illustration of the requirement is shown in Table 4.

Table 4 illustration of part of the requirements

Requirement documents for user: n
Context elements: Time [0-23],Distance(0-4)
Topic: root category [0-8],category[0-64]
Requirements: U_n_R1: The probability that user 49 has low requirements to participate in food activity in time = 9 is 0.381; U_n_R2: The probability that user 49 has high requirements to participate in food activity in time = 12 is 0.633; U_n_R3: The probability that user 49 has high requirements to participate in travel and transport activity in time = 18 is 0.719; U_n_R4: The probability that user 49 has high requirements to participate in shop and service activity in time = 20 is 0.417; … U_n_R34: The probability that user 49 has high requirements to participate in food activity in distance = 1 is 0.531; U_n_R34: The probability that user 49 has low requirements to participate in travel and transport activity in distance = 0 is 0.712; …

A tool is designed in order to facilitate the automation and generalization of the method. The tool including three parts, which are file upload and data analysis, role discovery and optimization. The functions of the first part are the construction of user features and the generation of user requirements. The function of the second part is user role discovery using k-means, DBSCAN or other algorithms. The functions of the third part are the optimization of the role discovery algorithm. Because the research content in this article only involves the first part, the remaining two parts will not be introduced in depth here.

The main interface of the tool is shown in figure 8. As shown in the figure, the buttons with three different colors in the upper right are the function buttons of the three parts of the system. The frame in the bottom left is used to display the result. Regarding the two boxes in the middle on the right, the one on the left is used to select context elements and topics, the one on the right is used to save of delete the result. The frame in the bottom right is used to show the process in running, and the requirement generated is also displayed in this framework.

The red buttons are used to implement the functions of the first part. The one in the right is Re-upload, which used to upload the data set. The six buttons in the middle including Density plots, Box plots, Histograms, Correlation matrix, Scatter Matrix and Heat map are used to calculate and display data and user features in different ways, such as Heat map and Histograms are shown in figure 9 and 10.

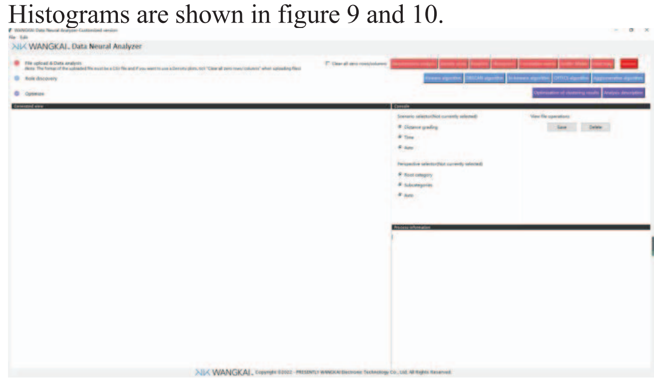

Figure 8. The main interface of the tool

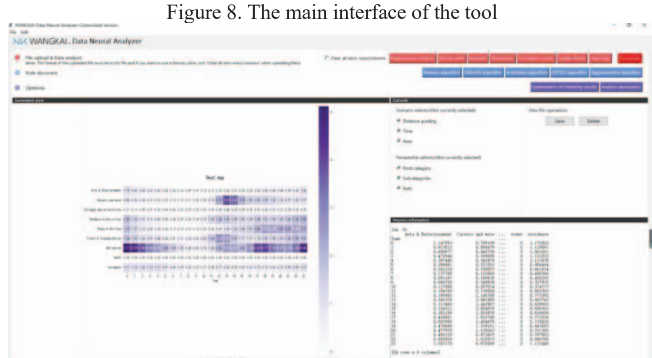

Figure 9. An illustration of Heat map of user feature

Figure 10. An illustration of Histograms of the data

The leftmost red button is requirements analysis, which is used to generate user requirements, a screenshot of the generation process is shown in the figure 11.

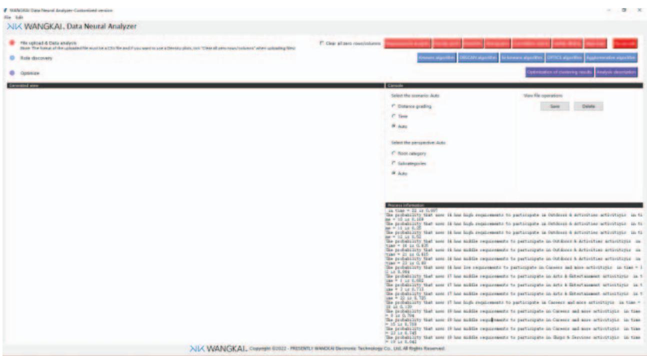

Figure 11. A screenshot of requirement generation process

V. CONCLUSION

How to discovery requirements from user data is challenging for developer of mobile application. In this study, user features are constructed based on user data, a time smoothing strategy is used to improve and accuracy of user's time related features, a fuzzy logic based method is proposed to generate user requirements, and a tool is developed to support the method.

Future works of the study includes the following aspects. Firstly, validate the method in larger datasets from various field, and extend the tool to support wider datasets. Secondly, the research in this paper is based on user check-in data, how to combine check-in data with data from various source such as user comments or scraped data to discover user requirements from more aspects is an interesting research direction.

ACKNOWLEDGMENT

This research was funded by National Natural Science Foundation of China, grant number No.62102291, No. U1504602 and No.61902447; this research was funded by University Key Scientific Research Project of Henan Province, grant number No.22A520052; this research was funded by High-level Talent Scientific Research Start-up Fund under Grant No. ZKNUC2021023.

REFERENCES

[1] 45nd Statistical Report on the Development of China's Internet, China Internet Information Center (CNNIC). http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/202202/t20220225_71727.htm. 2022,pp.16-25.

[2] Zheng Y. Location-based social networks: Users[M]//Computing with spatial trajectories. Springer, New York, NY, 2011: 243-276.

[3] Li D, Wong W E, Chau M, Pan S, Koh L S. A survey of NFC Mobile Payment: Challenges and solutions using blockchain and cryptocurrencies[C]. 2020 7th International Conference on Dependable Systems and Their Applications (DSA), 2020.

[4] Martin W, Sarro F, Jia Y. A Survey of App Store Analysis for Software Engineering[J]. Research Note of UCL Department of Computer,2016, pp.1-56.

[5] Li D, Wong W E, Pan S, Koh L S, Chau M. Design principles and best practices of Central Bank Digital Currency. International Journal of Performability Engineering[J]. 2021, 17(5): 411–421.

[6] Bernstein L, C. M. Yuhas. 2014. Software Requirements[M]. Apress, 73-106.

[7] Xu T, Ma Y, Wang Q et al. Cross-Urban Point-of-Interest Recommendation for Non-Natives[J]. International Journal of Web Services Research,2018,15(3):82-102.

[8] Xu C, Li D, Wong W E, Zhao M. Service caching strategy based on Edge Computing and Reinforcement Learning. International Journal of Performability Engineering[J]. 2022, 18(5): 350–358.

[9] Nieminen M P, Tyllinen M, Runonen M. Digital War Room for Design[C]//International Conference on Human Interface & the Management of Information: Information & Interaction for Learning. Springer-Verlag,2013.

[10] Pasquale L, Spoletini P, Pometto D, et al. Requirements engineering meets physiotherapy: An experience with motion-based games[M]//Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg,2013:315-330.

[11] Benabbou A, Bahloul S N, Dhaussy P. An Automated Transformation Approach for Requirement Specification[J]. Procedia Computer Science,2016,91:891-900.

[12] Gregoriades A, Hadjicosti J, Florides C, et al. A driving simulator for discovering requirements in complex systems[C]//Proceedings of the Conference on Summer Computer Simulation. Society for Computer Simulation International,2015.

[13] Niemann M. Elicitation of Requirements for an AI-Enhanced Comment Moderation Support System for Non-tech Media Companies[J]. 2021,pp.573-581.

[14] Li X , He Y , Peng Q , et al. A Computer-Aided Approach for Acquisition and Importance Ranking of Customer Requirements from the Online Comment Mining[J]. Computer-Aided Design and Applications, 2021, 19(1):132-151.

[15] Brill O, Knauss E. Structured and unobtrusive observation of anonymous users and their context for requirements elicitation[C]//IEEE 19th International Requirements Engineering Conference,2011.

[16] Xie H, Chang C K. Detection of new intentions from users using the CRF method for software service evolution in context-aware environments[C]//IEEE 39th Annual Computer Software and Applications Conference (COMPSAC). IEEE,2015.

[17] Hussein M, Yu J, Han J, et al. Scenario-driven development of context-aware adaptive web services[M]//Web Information Systems Engineering-WISE 2012. Springer Berlin Heidelberg, 2012:228-242.

[18] Hussein M Z B, Han J, Yu J, et al. Scenario-based validation of requirements for context-aware adaptive services[C]//2013 IEEE 20th International Conference on Web Services (ICWS). IEEE,2013.

[19] Kwon O, Lee Y. A structured methodology of scenario generation and system analysis for ubiquitous smart space development[M]//Ubiquitous Intelligence and Computing. Springer Berlin Heidelberg,2008:90-104.

[20] Alawairdhi M, Aleisa E. A scenario-based approach for requirements elicitation for software systems complying with the utilization of ubiquitous computing technologies[C]//IEEE 35th Annual Computer Software and Applications Conference Workshops. IEEE,2011.

[21] Maiden N, Seyff N, Grunbacher P, et al. Making mobile requirements engineering tools usable and useful[C]//14th IEEE International Requirements Engineering Conference(RE'06). IEEE,2006.

[22] Maiden N, Seyff N, Grünbacher P, et al. Determining stakeholder needs in the workplace: How mobile technologies can help[J]. Software, IEEE, 2007,24(2):46-52.

[23] Seyff N, Graf F, Grünbacher P, et al. Mobile discovery of requirements for context-aware systems[M]//Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg,2008:183-197.

[24] Jeong J , Lee Y K . Identifying Temporal Corpus for Enhanced User Comments Analysis[J]. International Journal of Software Engineering and Knowledge Engineering, 2022, 32(03):439-456.

[25] Liu Y, Liu L, Liu H, Yin X. App store mining for iterative domain analysis: Combine app descriptions with user reviews[J]. Softw: Pract Exper. 2019;49:1013–1040. https://doi.org/10.1002/spe.2693.

[26] Li B . Multi-classification of microblog's comments based on feature combination[C]// 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2021.

[27] Zhu J , Wickes E , Gallagher J R . A machine learning algorithm for sorting online comments via topic modeling[J]. Communication Design Quarterly Review, 2021,9(2):4-14.

[28] Lim K K , Lee C S . Sharing is Learning: Using Topic Modelling to Understand Online Comments Shared by Learners. 2021:91-101.

[29] Wang H, Liang T , Cheng Y . Prediction of Perceived Utility of Consumer Online Reviews Based on LSTM Neural Network[J]. Mobile Information Systems, 2021:1-7.

[30] Mantilla S C, Gutierrez J. Model Comparison for the Classification of Comments Containing Suicidal Traits from Redditvia NLP and Supervised Learning. Annual International Conference on Information Management and Big Data. Springer, Cham, 2022.

[31] Li Y. Construction and Adaptability Analysis of User's Preference Models Based on Check-in Data in LBSN[J]. 2021.

[32] Ya-li S, Fu-zhi Z, Wen-yuan L. A Time-aware POI Recommendation Method Exploiting User-based Collaborative Filtering and Location Popularity. Advanced Science and Industry Research Center[C]//Proceedings of 2nd International Conference on Communications, Information Management and Network Security (CIMNS),2017.

[33] Chen J, Zhang W, Zhang P, et al. Exploiting Spatial and Temporal for Point of Interest Recommendation[J]. Complexity,2018: 1-16.

[34] Yang D, Zhang D, Zheng V W, et al. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs[J]. IEEE Transactions on Systems, Man, and Cybernetics Systems, 2015, 45(1):129-142.

[35] Yang D , Zhang D , Qu B . Participatory cultural mapping based on collective behavior data in location-based social networks[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2016, 7(3):1-23.

[36] W.Hu, P J. Jin, et al. An adaptive hawkes process formulation for estimating time-of-day zonal trip arrivals with location-based social networking check-in data. Transportation Research Part C: Emerging Technologies,2017,79:pp.136-155.

[37] A. Noulas, S. Scellato, C. Mascolo. An Empirical Study of Geographic User Activity Patterns in Foursquare. international conference on weblogs and social media, 2011.

[38] M H. Kuo, L C. Chen, C W. Liang. Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. Expert Systems with Applications,2009,36(2), pp.3543-3554.