Yang, P

User Profiling in Elderly Healthcare Services in China: Scalper Detection

Article

Citation (please note it is advisable to refer to the publisher's version if you intend to cite from this work)

Yang, P (2018) User Profiling in Elderly Healthcare Services in China: Scalper Detection. IEEE Journal of Biomedical and Health Informatics, 22 (6). pp. 1796-1806. ISSN 2168-2194

User Profiling in Elderly Healthcare Services in China: Scalper Detection

Cheng Xie, *Member, IEEE,* Hongming Cai, *Senior Member, IEEE,* Yun Yang*, Lihong Jiang, *Member, IEEE,* and Po Yang

***Abstract*—Driven by the automation technologies and health informatics of Industry 4.0, hospitals in China have deployed a complete automation system/platform for healthcare services accessing. Without much more Internet knowledge, elderlies usually seek the third-party to assist them to get healthcare services from Web or APPs, it consequently results in an unexpected situation that scalpers could grab all healthcare services booking by unrighteous means in order to re-sell to elderlies for a much higher price. Moreover, it is hard for physicians to identify the scalpers due to the complexity, ad-hoc and multi-scenario nature of healthcare processes. In this paper, a novel method is proposed for the identification and creation of user groups of scalpers in mobile healthcare services. The approach utilizes and extends state-of-the-art data analysis approaches in the event-logs of the mobile system to identify user groups. Based on the user groups, user profiles are extracted by identifying representative event-cases from hierarchical user-event clusters. A comprehensive evaluation is conducted in a selected test-set from the event-logs of a mobile healthcare APP. The result shows its accuracy and effectiveness in scalper detection in mobile healthcare APP. Further, a complete case study is deployed in a real word hospital to ensure its utility, efficacy, and reliability.**

***Index Terms*—User Profiling, Mobile Healthcare, Scalper Detection, Elderly Services, Clustering, Process Mining**

I. Introduction

With the deep convergence of the automation technologies and health informatics driven by Industry 4.0, hospitals in China have deployed a complete automation system/platform for healthcare services accessing. Notably, by using automation booking device, mobile APP or authorized platform, doctor appointment services are directly accessed from the Internet. It has made the great convenience for the young patients who are skilled on the Internet. However, it is usually hard for chronic elderly patients who have to access healthcare services frequently but with the limited amount of Internet knowledge. Evermore, they even cannot obtain a healthcare service when scalpers are grabbing the services.

Since healthcare service is the urgently-needed resources in China, some "clever" users try to resale healthcare services by scalping in automation healthcare service. For example, some users use the mobile APP or the third party agent to store massive doctor appointments at one time at regular price. Then, they re-sell the appointments at a higher price to the elderly patients who did not get the service.

As shown in Fig. 1, the third part agents (scalpers) use robots or scripts quickly grab doctor appointments from the authorized platform, and then resell them to the normal users who can not obtain immediate appointments. In this case, some users would like to pay much higher price for urgent

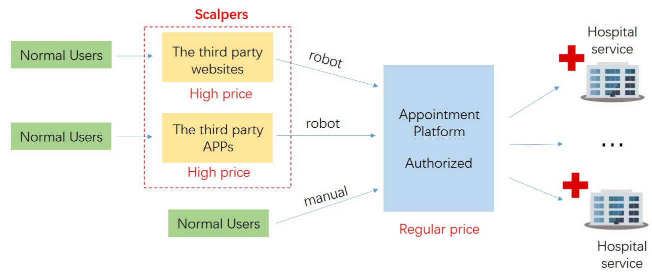

Fig. 1. The third party agents use robot, script or quick click to get hospital appointment from the authorized platform. Then, the agents ask much higher fee from the normal users who really need the service.

medical treatment. In general, scalping significantly consumes the hospital resources, and break the order of health service.

Both hospitals and users strongly suggest APP developers (Authorized Platform) control and reduce such scalpers in APP environment. A typical way to detect scalper is to analyze user's actions from event logs by domain experts. However, the event-logs of the APP are extremely large and quickly increased day by day (e.g., a healthcare APP for a hospital in Wuhan, China contains millions of events and increases about 400K events per week). It is difficult for domain experts to filter out scalpers from such data environment. Even well-trained APP administrators, to some extent, could not accurately distinguish scalpers from normal users when they sell the tickets (doctor appointments) offline. To optimize mobile APPs, support users with different goals and different levels of skills, and provide better user experiences, it is useful to identify and create user profiles (persona): representations of the goals and behaviors of a hypothesized group of users to filter out target users.

User profiles identify the user motivations, expectations, and goals that are derived from the online behaviors. System event-logs contain valuable information concerning user behaviors in mobile applications. Based on the analysis of user's event log, clustering algorithms could find out similar users based on their behaviors or interests and put them into groups. However, in general, healthcare processes are ad hoc [1], so it is difficult to filter massive noises. Moreover, due to the complexity and multi-scenario nature of users event sequences, it is a challenging task to effectively and accurately measure the similarity of user's event sequences.

Moreover, conventional methods for creating user profiles are manual. They are usually problematic because they are subjective, require the commitment of substantial resources, and rely on specialized skills.

In an attempt to address the drawbacks of the manual methods, according to the characters of healthcare processes, we propose an approach that integrates the user profiles identification and creation. First of all, according to the identification of the specific business scenario and users event log data, we create event case model based on the particular business scenarios. Then, the similarity of each event case is calculated by combining multiple matchers. After that, an extended hierarchical clustering algorithm is used to identify user groups based on event case similarity. Users that are judged to be similar to each other are grouped. Once the features of user groups are identified, profiling process extracts representative event cases from hierarchical user groups as profiles.

Finally, the discovered user profiles are applied to a detection process in a mobile healthcare APP to identify ticket scalpers. In short, we make the following contributions:

- A specific scalper detection framework is proposed for elderlies in use of mobile healthcare services.
- A novel clustering-based approach is developed for discovering user profiles from APP event-logs.
- Practical test shows that our approach works well with real-world healthcare APPs in a hospital, it significantly helps administrator to identify scalpers from complex data environment.

The rest of the paper is organized as follows: Section II provides the overview of the proposed framework. Section III describes the core methods of user profiling and scalper detection in details; Section IV presents the case study of scalper detection by using the framework in a real-world healthcare APP; Section V summarizes and compares the related researchers and methods; Section VI concludes the works.

II. Overview of the Framework

Fig.1 gives the overview of the framework. The inputs of the framework are the event-logs from mobile App. The outputs of the framework are the detected scalpers with their profiles. The framework consists of four processes that are data modeling, EventCase similarity calculation, EventCases clustering, user profiling and scalper detection, as shown in Fig.2.

- Data Modeling is used to format user, event and process (EventCase) from system event logs into a unified model.
- EventCase Similarity Calculation. In this step, based on the data model, the similarities between processes are calculated. A pair-to-pair similarity matrix of processes is created.
- EventCases Clustering. Based on the similarity matrix, an agglomerative hierarchical clustering is applied to group the processes into hierarchical clusters with labels C_1, C_1,...C_n.
- User Profiling. According to the clusters of processes, users are then grouped by their related labels of process clusters. By analyzing the center user of each group, the representative process of the center user could be found as the profile of the group.

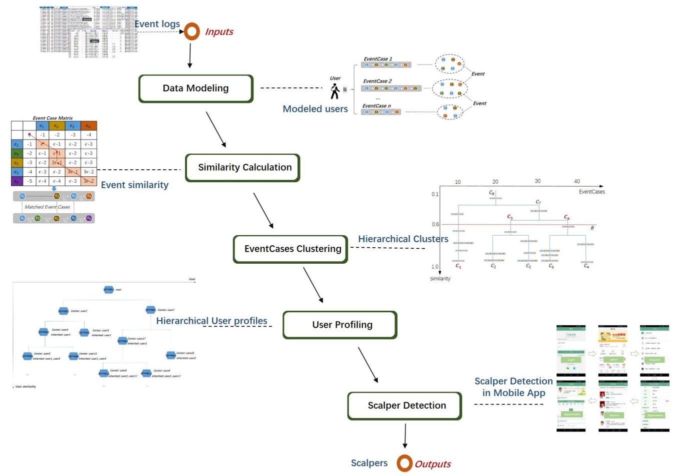

Fig. 2. The overview of the framework.

- Scalper Detection. Based on the profiles of the user group, scalpers are detected by filtering users from combined profiles.

In the following sections, step-by-step description of the user profile and scalper detection will be introduced. According to the framework, Section III.A gives the definitions of the user, event and process model. Similarity calculation of process is discussed in Section III.B. Process clustering and User profiling are introduced from Section III.C to III.D. Scalper Detection is then described with a real-world application in section IV by using user profiles.

III. Method

In this section, we describe the proposed method for the identification and creation of persona in detail.

A. Data Modeling

The server log records events, which represent activities and associate with particular event cases. Each event case can be represented by a sequence of events. Event logs that are recorded by information systems are usually too redundant and unstructured. And event cases are usually hard to be extracted. In this paper, we apply the business scenario driven analysis method to extract event cases for persona description.

To reason about logs and to precisely specify the requirements for event logs, we formalize the various notions.

Definition 1: An **User** is an information entity that represents person(patient) who uses the information system. It consists of an unique identifier UID and a set of attribute:

$$User = (UID, \{Attr_1,\ Attr_2\ ...\ Attr_n\}) \tag{1}$$

- UID (User ID) is an identifier for a User.
- $\{$Attr$\}$ is a set of attribute that belongs to the User.

For example, in the case study, the attributes of a user include “gender”, “brithday”, “register_date”, “phone_number”, “medical_guide”, etc.

Definition 2: An **Event** is an abstracted concept that represents when/where/who an activity is related to. In particular, an event can be a button-click of select, submit, search or other items in an information system:

$$Event = (EID, UID, A, L, T) \tag{2}$$

- EID (Event ID) is an identifier for an Event.
- UID (User ID) is an identifier for a User.
- A is an action that the user performed. The name of action is pre-defined in the event-log system.
- L is the location where the action occurred. In Mobile Healthcare, the location is the GPS coordinates from mobile APP denoting in (x,y).
- T is the time when the action happened. Normally, the time is the server time.

Definition 3: An **EventCase** is a process that a user performed to finish a business. It can be "make an appointment", "search a Chief Physician" or other businesses in a system. Normally, EventCase is extracted and modeled from information system log such as event-log of HIS (Health Information System) and MHS (Mobile Healthcare System) :

$$EventCase = (CID, \{e_1,\ e_2\ ...\ e_n\}) \tag{3}$$

- CID (Case ID) is an identifier for an EventCase.
- $\{e_n\}$ is an ordered list of event that belong to the EventCase. e_n represents the n-th event in an EventCase."e_0, e_1, , e_n" becomes an event sequences.

According to the definitions, the relationships among $User$, $Event$ and $EventCase$ are described in fig.3.

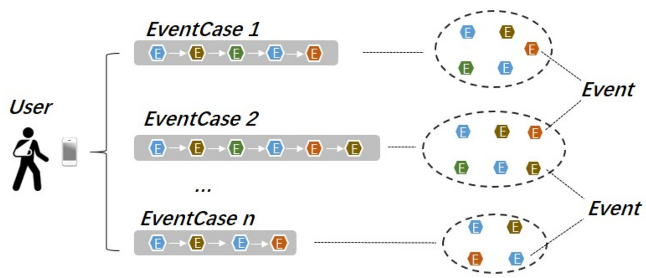

Fig. 3. The relationships of Event, EventCase and User model.

In the model, a $User$ could have more than one $Even-Cases$ which represent different process the user performed such as "making an appointment" or "searching a doctor" in the information system. Similarly, an $EventCase$ contains more than one $Event$. $Events$ are combined orderly to became an $EventCase$.

B. EventCase Similarity

Activity Matcher (AM) is a wordnet-based string matcher. It calculates the similarity between two words (word expressions) in the activity of an event by using Information Content(IC)[2] in a wordnet graph:

$$AM(e_1.A, e_2.A) = \frac{2 \cdot IC(LCS(e_1.A, e_2.A))}{IC(e_1.A) + IC(e_2.A)} \tag{4}$$

Here, e_1 and e_2 are the events while $e_1.A$ and $e_2.A$ are the activities of events. Wordnet IC can be downloaded from WN-Similarity[1] and LCS is "Lowest Common Subsumer" that represents the closest superclass of w_1 and w_2 in wordnet taxonomy.

Location Matcher (LM) is based on coordinate matching algorithm. It matches longitude and latitude between two locations

$$LM(e_1, e_2) = \frac{L_1.x \cdot L_2.x + L_1.y \cdot L_2.y}{\sqrt{L_1.x^2 + L_2.x^2} + \sqrt{L_1.y^2 + L_2.y^2}} \tag{5}$$

Here, L is a location where L.x is the longitude and L.y is the latitude of the location. The output of the LM is the cosine similarity of two locations.

Sequence Matcher (SM) is an AM, LM and time combined ordered sequence matching algorithm. It extends Needleman and Wunsch[3] matching approach to match sequence by using AM and LM to match the single node of the sequence. Here, sequence (EventCase) consists of ordered events. An example of EventCase matching by using matchers are provided in Fig.4.

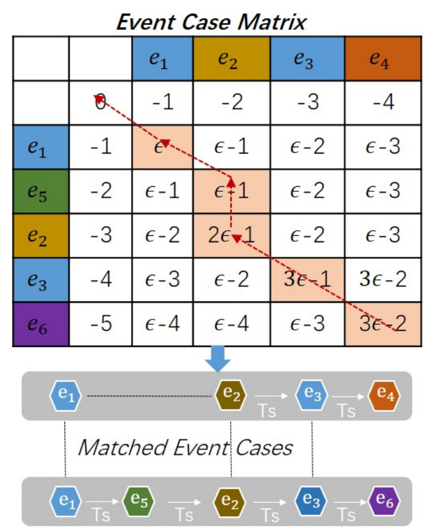

Fig. 4. An example of EventCase matching.

In the figure, the process and parameters (CaseMatrix and ϵ) of EventCase matching will be formalized as follows:

Definition 4: A **CaseMatrix** is a matrix of similarity scores for any two EventCases (sequences). It can be described in a two-dimensional array:

$$CaseMatrix = CM(i, j) \tag{6}$$

Here, i is the index of an event in one EventCase and j is the index of an event in another EventCase. For each mismatch and indel event, we get -1 score while +ϵ for the matched event. Algorithm 1 provides the detail process of CaseMatrix construction.

Algorithm 1 combines Activity Matcher, Location Matcher and timestamps of an event to build a CaseMatrix CM(i,j) for any two EventCases. Based on CaseMatrix we can easily

[1]http://www.d.umn.edu/∼tpederse/Data/WordNet-InfoContent-3.0.tar.gz

Algorithm 1 CaseMatrix construction

Input: Two EventCases ec_1 and ec_2
Output: CM(i,j)

for i=0 to length(ec_1) **do**
  $CM(i,0) \leftarrow -i$
end for
for i=0 to length(ec_2) **do**
  $CM(0,j) \leftarrow -i$
end for
for i=1 to length(ec_1)) **do**
  for j=1 to length(ec_2)) **do**
    # Combining with Activity Matcher:
    As= $AM(ec_1(i), ec_2(j))$
    # Combining with Timestamp:
    Ts=||$ec_1(i)$.T-ec_1(i-1).T|-|$ec_2(j)$.T-c_2(j-1).T||
    # Combining with Location Matcher:
    Ls= $LM(ec_1(i)$, $ec_2(j))$
    $\epsilon(i,j) = \frac{Ls \cdot As}{Ts}$
    Match $\leftarrow$ CM(i-1,j-1) + $\epsilon(i,j)$
    Mismatch $\leftarrow$ CM(i-1, j) - 1
    Indel $\leftarrow$ CM(i, j-1) - 1
    CM(i,j) $\leftarrow$ max(Match, Mismatch)
  end for
end for

calculate the similarity scores between EventCases by using Algorithm 2.

Algorithm 2 Similarity Calculation

Input: Two EventCases ec_1 and ec_2; CM(i,j)
Output: Similarity score sim

i $\leftarrow$ length(ec_1)
j $\leftarrow$ length(ec_2)
while i $>$ 0 or j $>$ 0 **do**
  if i $>$ 0 and j $>$ 0 and CM(i,j)=CM(i-1,j-1) + $\epsilon(i,j)$ **then**
    sim $\leftarrow$ sim+$\epsilon(i,j)$
    i $\leftarrow$ i - 1
    j $\leftarrow$ j - 1
  else if i $>$ 0 and CM(i,j)=CM(i-1,j) -1 **then**
    sim $\leftarrow$ sim - 1
    i $\leftarrow$ i - 1
  else
    sim $\leftarrow$ sim - 1
    j $\leftarrow$ j - 1
  end if
end while
return Normalized(sim)

After the calculation in Algorithm 2, we could obtain matched EventCases like we illustrated in Fig.2. To note that, in the algorithm, similarity sim is normalized with the maximum score of matched EventCase pair.

C. EventCase Clustering

Based on EventCase similarity calculation, we apply Agglomerative Hierarchical Clustering (AHC), which is a similarity-based hierarchical clustering[5], [6], to build EventCase clusters. An example of hierarchical EventCase clusters is illustrated in Fig. 5. AHC is a "bottom-up" approach, which means that each node starts out as a single cluster. Then pairs of clusters are combined into larger ones as the process continues until only one cluster is left[4]. Combing with EventCase similarity calculation (Algorithm 1-2), Algorithm 3 takes the EventCases and a similarity as input, AHC as a procedure, to build hierarchical clusters for EventCase.

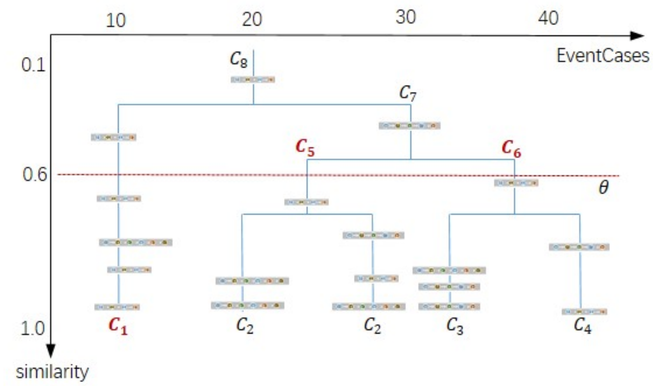

Fig. 5. An example of Similarity-Based AHC clustering for EventCases.

Algorithm 3 Similarity-Based AHC clustering for EventCases

Input: Similiarty Matrix SM(i,j), EventCases ECs.
Output: Clusters $\{C_1, C_2 ... C_n\}$;

while $|ECs| > 0$ **do**
  for i= 0 to $|ECs|$ **do**
    for j= i to $|ECs|$ **do**
      sim $\leftarrow$ SM(i,j)
      if $sim > Max$ **then**
        Max-sim $\leftarrow$ sim
        Max-pair $\leftarrow$ (i,j)
      end if
    end for
  end for
  C_{new} $\leftarrow$ merge(Max-pair)
  C.add(C_{new})
  ECs.add(C_{new})
  ECs.remove(Max-pair)
  update(SM) # using group averaging
end while
return C

In the algorithm, in each iteration, the function "update" is used to average similarities of EventCases in the new cluster. In the next iteration, the updated similarity score will be used for the cluster. The output of the algorithm is a set of EventCase cluster. It then will be used for user clustering in the next section.

D. User Profiling

In Section III.C, we have built EventCase clusters C. In each cluster $C[i]$, there are EventCases which are related to corresponding users by EID and UID (see Definition 2). Thus, giving a User u, its related clusters $\{C_1...C_m\}$ can be also found. Then, by counting the number of user's EventCases contained in clusters, the user attributes (cluster, number) are created. Fig. 6 gives an example of cluster-number attributes for a user (Definition of User and attributes are provided in Equation 1).

User

UID	C_1	C_2	...	C_m
11089644	7	2	...	4
11145118	0	6	...	2
11085559	8	3	...	5
...	...	...		...

Fig. 6. An example of User-attributes, cluster and counting numbers.

User similarity. Based on User-Attributes, the similarity scores between users could be calculated by using equation (7).

$$Sim_u(u_1, u_2) = \frac{u_1.C_1 \cdot u_2.C_1 + \cdots + u_1.C_m \cdot u_2.C_m}{\sqrt{u_1.C_1^2 + u_2.C_1^2} + \sqrt{u_1.C_m^2 + u_2.C_m^2}} \tag{7}$$

Profile discovering. The idea of user profiling is to group (clustering) users from top to bottom. For each group, there is a representative EventCase represents the feature (profile) of the group while the bottom group holds all features (profiles) from its upper groups. Here, based on User similar Sim_u as distance function, we apply Divisive Hierarchical Clustering (DHC)[7] to build top-bottom user groups to find user profiles:

- Step 1: all users are assigned into root group.
- Step 2: based on user similarity, 2-means clustering is used to divide the group into two groups.
- Step 3: recording center points (profile) of 2-means clustering for the two groups.
- Step 4: repeating step 2-3 utill all users are assigned into separated (bottom) groups.
- Step 5: using center points (profile) to annotate each group.

Figure 5 gives an example of user profiling in the mobile APP of doctor appointment.

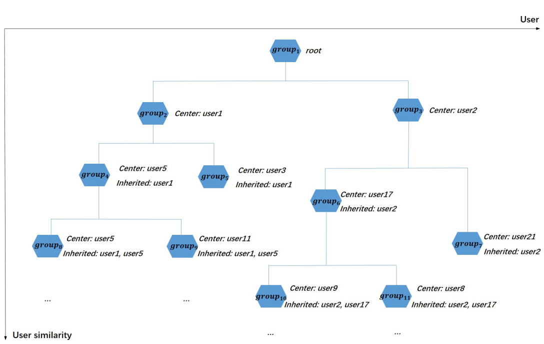

Fig. 7. An example of user profiling in the mobile APP of doctor appointment.

After user grouping, an administrator could obtain all center users for each group. Then, according to the particular business, e.g. hospital appointment, an administrator could easily profile these center users by checking their representative EventCase (e.g. "appointment applied but canceled", "fast EventCase performed", "appointment succeed but no review on doctors" and so on).

IV. Case Study

In this case, we studied our approach in a real-world mobile healthcare APP.

A. Scenario Description

Ticket scalpers (also called Huang Niu in China) are the most influential but hard to be detected users in mobile APP of the doctor appointment. Both hospitals and APP developers are suffered from Huang Niu who rushes to take almost all the appointments from normal users. Qu Yi Yuan[2], our collaborative healthcare software company, is one of the APP companies suffered from Huang Niu. By using Qu Yi Yuan APP, patients could easily get appointments from almost all the hospitals in Shanghai without going out their home. Such convenience, of course, is for both patients and Huang Nius. A simple event-process of appointment in Qu Yi Yuan APP is shown in Fig. 8.

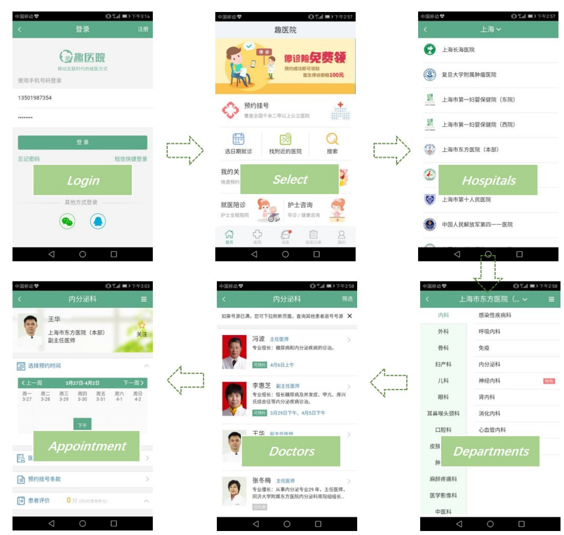

Fig. 8. A simple event-process of appointment in Qu Yi Yuan APP.

The APP provider Qu Yi Yuan provides us event-logs to try to find Huang Nius out of normal users. The core events of Huang Nius are very similar to the events that normal users did since both of them are willing to succeed the appointments. However, the purpose of Huang Nius (to sell the tickets) is much different from normal users(to see the doctors), i.e., Huang Nius only concern to get appointments successful. Thus, it is believed there are some differences hidden in user's event-log could be found for Huang Niu detection.

[2]https://www.quyiyuan.com/

B. Dataset Description

Qu Yi Yuan provides us a user event-log of a particular hospital in Wuhan, China. The log contains 5,906 users and 398,764 events. Both Qu Yi Yuan and related hospitals want to know how many users in this dataset might be Huang Niu. A fragment of event-log is shown in Fig. 9.

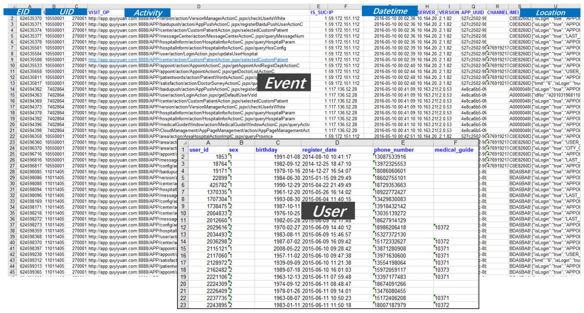

Fig. 9. A fragment of system event-log of Qu Yi Yuan mobile APP.

The event-log consists of user records and event records. User records contain name, idCard, account, password, medical guide, birthday, register date, phone, and gender. Event records include the event URL a user has clicked, when and where the user clicked the URL, and other related information. Further statistic features of the event-log are given in Table I.

TABLE I
DESCRIPTIVE STATISTICS OF EVENT LOG FROM QU YI YUAN APP

Event	Total	398,764	Distinct	204
	Top 10 Events			
	Event Name		Clicked	Proportion
	../selectedCustomPatient		40,984	10.2%
	../queryDoctorCareInfo		14,004	3.5%
	../getDoctorListAction		13,917	3.5%
	../queryDoctorSatisfactionRecord		13,183	3.3%
	../checkUserIsWhite		12,833	3.2%
	../registerBaiduPushUserAction		12,705	3.2%
	../queryCity		10,107	2.5%
	../queryProvince		10,053	2.5%
	../appointRegistResultAction		9,499	2.4%
	../getAppointAndRegistDeptAction		8,004	2.0%
User	Total		5906	
	Male	1,790	Proportion	30.3%
	Female	3,548	Proportion	60.0%
	Unknown	568	Proportion	9.7%
	Age: 0-30	1,127	Proportion	19.1%
	Age: 31-60	2,747	Proportion	46.5%
	Age: 61+	1,464	Proportion	24.8%
	Unknown	568	Proportion	9.7%

Besides, Qu Yi Yuan also provide a test-set with 120 users and 8,720 events. In the test-set, administrators of the APP have manually checked every user with their events. There 22 users are marked as Huang Niu and the rest 98 users are normal users. The test set is used to evaluate and set up the parameters of the approach.

More importantly, before Qu Yi Yuan provides us the data, they have conducted a data masking process. The data applied in the work is only for the research purpose. The data masking process is conducted on user records by masking name, account, password, and idCard. The detail of the data masking process is described in Table II. Further, the research has also been reviewed by the Institutional Review Board of the school of software, Yunnan University and also approved by the APP provider.

TABLE II
THE DATA MASKING PROCESS CONDUCTED BY THE APP PROVIDER

Column	Before masking (example)	After masking (example)
name	Yueming Zhang	Z********g
idCard	53270119810807223X	AUTO_INCREMENT_ID
account	zhangym7765	z******5
password	C4CA4238A0B...	*********
medical_guide	wangxing1988	AUTO_INCREMENT_ID
birthday	1982-9-12	1982-9-12
register_date	2014-08-10 10:41:17	2014-08-10 10:41:17
phone_number	13087533916	13087533916
sex	1	MALE

C. Data Modeling

Data Modeling is used to model user, event, and eventCase together to prepare for eventCase clustering. First, we format user and event log by using Definition 1 and 2. Then, we apply proM[3] to find EventCase (Definition 3) from user and event log. After process mining, the major EventCases of doctor appointment are modeled, as showed in Fig. 10.

From the EventCases, we can see a user could succeed an appointment in different ways. Some users directly go to the last pages to finish the appointment. Some users are willing to detailed view the doctor information before the appointment. Some users find the doctor from their previous records or treatment. Other users perform an appointment depend on their patient card in a particular hospital.

D. EventCase Clustering

According to Section III.B and III.C, EventCase Clustering is used to build user features depending on the EventCase clusters the user is related. EventCase represents real activities an user performed in APP that implies normal/abnormal users. By using Algorithm 1-3, the clusters of EventCases in test-set are created, as showed in Fig. 11.

In Fig. 11, the threshold θ is used to cut the hierarchical clusters to obtain proper clusters. The set-up of this parameter will be discussed later. In this case, when the distance greater than 0.853, there is only one cluster left. When the distance is closed to zero, all EventCases become separated clusters. Here, θ is set as 0.5 to cut the hierarchical clusters, 10 clusters are obtained. Then, these clusters are labeled as $C_1, C_2...C_{10}$ and as the attribute set of users.

E. User Profiling

Based on EventCase cluster $C_1, C_2...C_{10}$, we apply DHC clustering (see Section 3.4) on 120 users in test-set to build top-bottom user groups. The result of user groups is shown in Fig. 12.

From the result shown in Fig. 12, 17 groups from 120 users in test-set are found. UID in each group is the center user

[3]https://sourceforge.net/projects/prom/

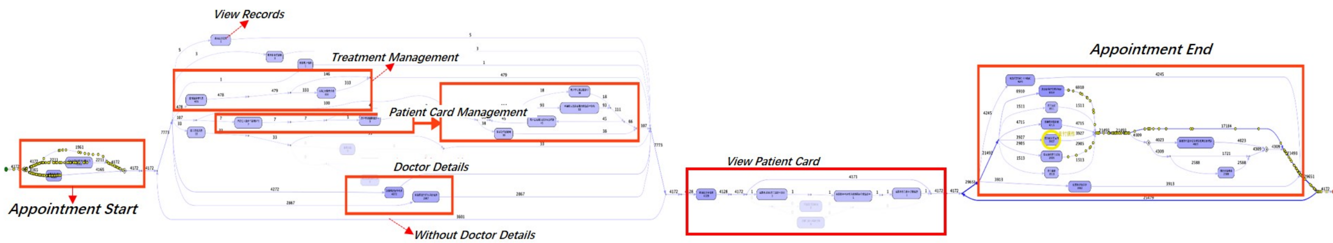

Fig. 10. The major EventCases of doctor appointment mined from Qu Yi Yuan APP.

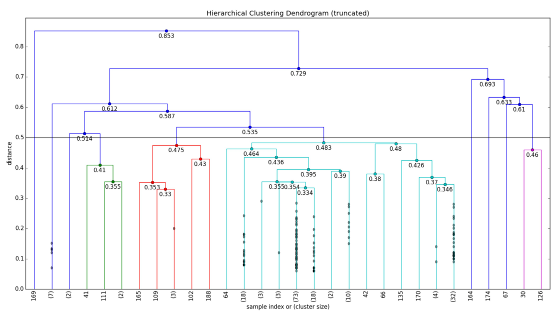

Fig. 11. The clusters of EventCases in test-set. Here, distance= 1- similarity. Since the whole clusters are too big to explore, we have truncated the clusters when the distance is shorter than 0.01

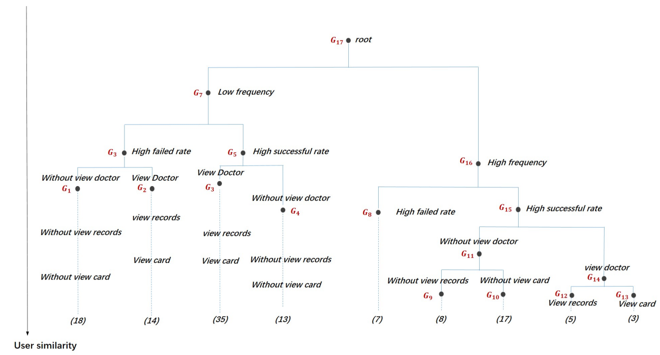

Fig. 13. User profiling by replacing center UID by its representative activities. It is worthing to note that the Representative activities are required manual analyzing on user's EventCase according to different business requirements.

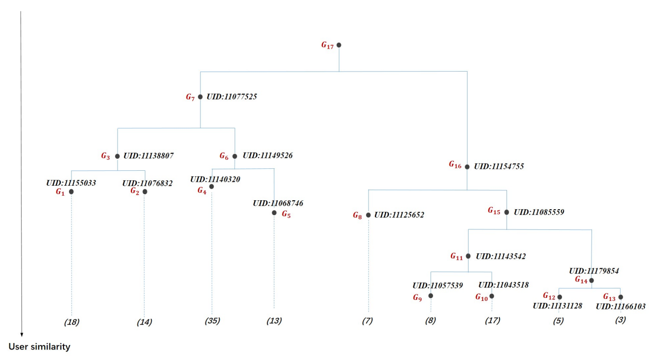

Fig. 12. The user groups based on EventCase clusters. Here, we found 17 groups. UID in each group is the center user of the group. The bottom groups hold all features from the upper groups.

of the group. The bottom groups hold all features from the upper groups. Group G_7 holds 80 users and G_{16} holds the rest 40 users. This might imply two different groups of users in using the APP in different ways. Based on the center UID of each group, we manually check all the center users and select their representative activities to replace the UIDs, e.g., the representative activities could be "high frequency in doctor appointment", "high failed rate", "without checking patient card" etc. Then, we get the following user profiles.

In Fig. 13, each group holds at least one profile. The bottom groups hold all profiles of their upper groups. For example, group G_9 has a profile "Without viewing records". It also has the profiles "Without viewing doctor","High successful rate" and "High frequency".

F. Scalper Detection

Based on profiled user groups, we start scalper (Huang Niu) detection on test-set. In the test-set, all Huang Nius are already marked. The evaluation is to compare Huang Nius discovered by the approach with the marked Huang Nius. The evaluation metric is the standard Precision (P), Recall (R) and F1-measure (F) metric. P = (True Positive)/ (True Positive + False Positive), R = (True Positive) / (True Positive + False Negative), F = 2 * P * R/(P + R). We first assume all users in G_1 are Huang Niu and calculate F1-measure. Then, we try G_2, G_3...to G_{17} to see which group achieves the highest F1-measure. The result is showed in Figure 13.

From the result showed in Fig. 14, scalper detection achieves the highest F1-measure (0.77) while group G_{11} (The profiles are: "high frequency", "high successful rate" and "without view doctor") is selected. 18 correct Huang Nius are found but 4 are false positives. At this time, the precision achieves 0.72 with 0.82 on the recall. The detection achieves the highest precision (0.88) while group G_9 (The profiles are : "high frequency", "high successful rate", "without view doctor" and "without view records") is selected. 7 correct Huang Nius are found with one false positive. The precision achieves 0.88 but with significantly decreasing on the recall (0.32). The detection achieves the highest recall (1.0) while group G_{17} (The profile is: "root") is selected. 22 correct Huang Nius are found but with 98 false positives.

From the Huang Niu distribution shown in Fig. 15 (a), we find almost all the Huang Nius are included in group G_{10}, G_{11}, G_{15}, G_{16} and G_{17} which are indeed in the same hierarchy (the right side hierarchy in Figure 12). It implies the group G_{16}

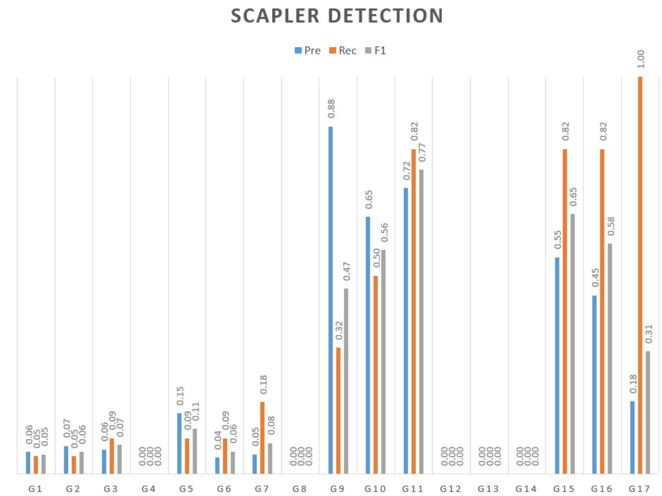

Fig. 14. The result of scalper detection on test-set of 120 users. The blue bar is precision while the orange bar is recall and gray bar is F1-measure.

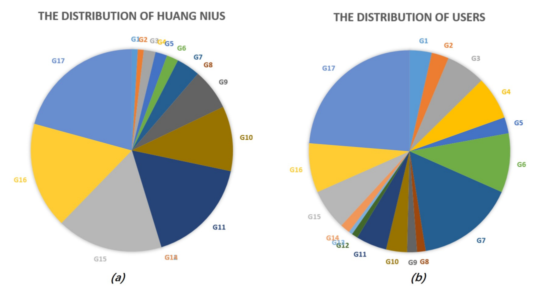

Fig. 15. (a) The distributions of Huang Nius and (b) users in each group.

(the root of the right side hierarchy) with the profile "high frequency" determines the most Huang Niu in the detection. It also fits the features of Huang Niu in reality who try to get more doctor appointments every day. Fig. 15 (b) shows the most of all normal users are gathered in group G_3, G_3 and G_7 which belong to the same hierarchy (the left side hierarchy in Figure 12). It shows normal users tend to get single or fewer appointments at one time. But we also find there are 2 Huang Nius in G_4("low frequency", "high successful rate" and "without view doctor") which are the subset of G_7 ("low frequency"). Though these Huang Nius do not perform many appointments at one time, they are very skilled in getting an appointment without caring about their doctors. These could be another feature of Huang Nius. Our approach did not detect these Huang Nius since in the groups of these Huang Nius there are still many normal users (Some normal users might familiar with APP so that get a relative high successful rate on appointment). It requires more information to distinguish such Huang Nius out from normal users.

From the evaluation on test-set, the proper set-ups could be selected. Here, the profiles for Scalper Detection are selected from group G_{11}, i.e., "high frequency", "high successful rate" and "without view doctor". The similarity threshold θ for EventCase clustering is set as 0.5. Then, we apply the same process on the hospital dataset to detect unknown Huang Nius. The result is showed in Fig. 16.

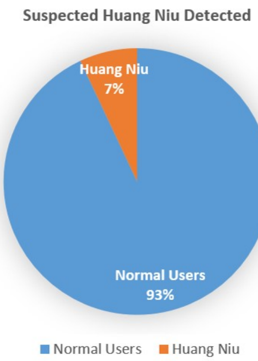

Fig. 16. The suspected Huang Nius are detected from 5907 users in a particular hospital. In total, the approach discovers 413 suspected Huang Nius out of 5907 users.

From the result showed in Fig. 16, out of 5097 users, there are 413 suspected Huang Nius are detected. According to the precision 0.72 of the approach in test-set, it might be more than 100 suspected Huang Nius are false positives. In the real hospital users, it is hard to build a gold standard in advance. However, experts of software company of the APP suggest us to randomly check 30-50 suspected Huang Nius manually since they are more concern about accuracy than recall. Indeed, it is essential to guarantee high accuracy in Huang Niu detection due to the UE (User Experience) requirement. Thus, we randomly check 50 detected Huang Nius by human experts to evaluate the accuracy. The result is showed in Fig. 17.

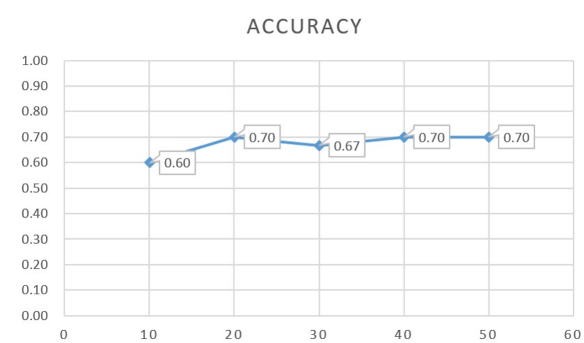

Fig. 17. The accuracy of scalper detection in the dataset of 5907 users from a particular hospital by using profile of G_{11}. In total, the approach discovers 413 suspected Huang Nius out of 5907 users.

From the evaluation, we find that 124 out of 413 suspected Huang Nius are more likely to be normal users. For example, though some users conduct many appointments without viewing doctor details in a few days. They actually appoint doctors for their children, families or related persons that can be found in patient records through manually checking. It implies these users are more likely to be normal users. Thus, according to UE requirement, we restrict profiles from G_{11}(the highest F1-measure) to G_9 (the highest precision with more restricted profiles). The result is showed in Fig. 18.

In Fig. 18, the result shows the detection has achieved high accuracy (0.9+). Though the recall is obviously decreased, it already fits the requirement of the APP developer (trade-off

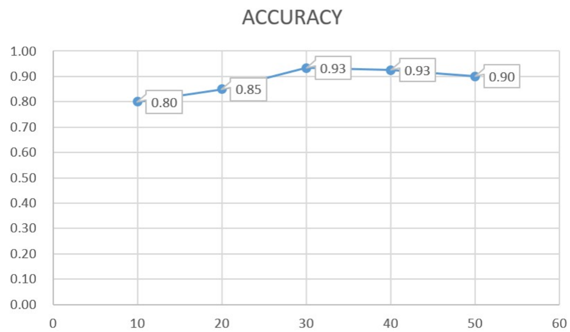

Fig. 18. The accuracy of scalper detection in the dataset of 5907 users from a particular hospital by using profile of G_9. In total, the approach discovers 320 suspected Huang Nius out of 5907 users.

between accuracy and recall). In this case, out of 5907 users, 320 suspected Huang Nius are detected with the corresponding user profiles “high frequency”, “high successful rate”, “without view doctor” and “without view records”.

G. Analysis of the Experiment

Almost all the processes of the approach are automated except representative event case analysis (to select a profile for a center user in the particular user group). However, depending on the particular requirements, e.g., to find Huang Niu from a particular APP, an automation system could also be developed to select a profile from a predefined label set.

The proposed approach does not achieve the highest precision with the highest F1-measure. But the approach is flexible to adjust profiles to reach a relatively high accuracy with a little lost in the recall. It is very useful in high UE required application such as Qu Yi Yuan for the doctor appointment.

The proposed approach is applied on a real hospital data environment to help APP administrators to detect scalpers. It detects 320 suspected scalpers out of 5907 APP users with estimated 0.9+ accuracy. Based on the idea of user profiling of the approach, it could also be applied to other APPs, social networks or online gamer to detect robot scalpers.

V. Related Works

According to the methods introduced above, first, related works are discussed from similarity calculation on an event to event/user clustering and user profiling. Then, for the application, state-of-the-art healthcare works are also discussed. At last, we summarize both the related and our works.

A. Similarity Calculation

As we introduced above, similarities are the key elements to identify the relationships between processes and users. There are already many similarity calculation methods existing in the domain and could be roughly divided into two categories: non-semantic similarity and semantic similarity. Non-semantic similarity calculation is, somehow, string-based literal matching, including word matching, numeric matching, date matching, string matching and other string related matching. In this domain, Bizer and his team [19] have proposed an effective and complete similarity calculation, so-called matchers, to calculate different data objects. Such matchers could be string matcher, date matcher, location matcher, name matcher or other string-based matchers. In the other way, semantic similarity calculation is, normally, link (or edge)-based graph matching. The idea of such approaches is that two nodes are similar if their neighbors are already similar. Thus, in a graph, the semantic similarity of two nodes are actually calculated from their neighbors, not from themselves (i.e., neighbors' similarities are calculated from neighbors' neighbors). The representative approaches of semantic similarity calculation could be page-ranking [20], Simrank [21], Similarity-Flooding [22] and semantic-graph-similarity [23], [24], [25]. It is difficult to say which is the better between semantic and non-semantic approaches. Depending on particular requirements, one could decide which methods to use. In the paper, we compare string matcher, date time matcher, location matcher, and wordnet-based string matcher into sequence matcher to resolve similarity calculation between processes.

B. Related Clustering Methods

Both process and user grouping in our approach relate to similarity-based (or say distance based) hierarchical clustering. In particular, AHC is applied on process clustering while DHC on user grouping.

One possible approach for hierarchical clustering is bottom-up. Initially, each item is put into its own cluster, and on each iteration, two clusters are selected and merged into a larger one. This approach is often called agglomerative, but the algorithm is known by many names, such as Globally Closest Pair (GCP) clustering [27], Sequential Agglomerative Hierarchical Non-overlapping (SAHN) clustering [28], [29] or Agglomerative Hierarchical Clustering (AHC) [30], [31]. For the event sequence data, temporal data clustering approaches are needed. HMM-based clustering [32], [33], [34] for temporal data could be applied on AHC to import temporal data into hierarchical clustering. Our approach combines AHC and temporal event sequences to create hierarchical process clusters.

The other possible approach is up-bottom. Initially, all the items are put into root cluster. Then, the cluster is split into two clusters by the center item based on the similarity matrix . The splitting could be stopped while each item becomes a separated cluster. For different problems, up-bottom hierarchical clustering has different specializations. C. sarbu with his team [26] use Gustafson-Kessel algorithm to implement a Fuzzy-DHC clustering to group soil samples in the chemical domain. T Xiong et al.[35], [36] use multiple correspondence analysis (MCA) to implement a novel DHC clustering for categorical data. Our approach, based on process clusters, combines original DHC with K-means clustering to implement a specialized DHC to build hierarchical user groups.

C. Profiling

User profiling is well-known in many usages such as using profiles to predict user's preferences (so-called preference elicitation) [40] on movies, using web browsing history to profile users to query relevant web pages [37], using user profiles to boost query keywords to get interesting results [41], using user profiling approaches for demographic recommender systems [38], etc. There is no the best approach for user profiling. According to different usages (business requirements), different profiling methods are selected. In details, [40] uses discovered preference rules (e.g., War⇒ Sport: 41%) as user profiles; [37] uses the browsing url (Web url the users have browsed) as profiles; [38] uses demographic attributes (such as age, gender, education, etc) of persons as profiles to categorize users; [41] uses shared items (could be a post/url/document/others a user shares) as user profiles. In our approach, according to the particular environment of app users, the representative events of APP users are selected as profiles.

D. Mobile Healthcare System

Mobile healthcare systems (MHSs) have successfully addressed many healthcare issues related to clinical decision support such as for field health workers information assistance [9], [10], for cardiovascular disease [13], for rural communities healthcare [8], incorporating patient data streams [14], offering epidemiological support for managing infectious disease [15] etc. Currently, healthcare organizations are shifting from paper-based record systems to HIS (Health Information System) systems, especially mobile healthcare system which collects Lifelogging from wearable or mobile devices[11], [12], so as to improve the quality of the provided care. In fact, it is a common belief, also supported by evidence [16], [17], that computer-based communication has positive effects on improving healthcare efficiency, safety, reducing costs and is better than existing communications means such as the postal service or hand-delivery [18]. Various MHSs are becoming the most welcomed tools for patients and physician in communication, diagnosis, treatment and other activities. This paper is focused on a mobile healthcare system dealing with doctor appointment which is believed as one of the most common transactions between patient and physician.

E. Summary

In the summary, our approach combines and extends the state-of-the-art data processing technologies to solve the real world challenges (scalpers in healthcare) existing in a mobile information system. Table III summarizes the core technologies in the approach.

From the comparison described in Table 2, our approach did not invent novel similarity calculations nor clustering algorithms in profiling. But our approach compares and extends existing state-of-the-art methods to propose a novel method on user profiling especially in the domain of mobile healthcare.

VI. Conclusion and future work

Automation technologies and health informatics, to a certain extent in China, increase the possibility of influencing healthcare resource distribution via scalping. In this paper, we proposed a method for mining user profiles from event logs of mobile healthcare APP to detect ticket scalpers in elderly healthcare services. A set of experiments on a real-world test set and hospital event-logs showed the efficiency of the method. The method was then deployed in a healthcare APP, called Qu Yi Yuan, to analyze ticket scalpers. It achieved 72% precision and 77% recall for scalper detection on the test dataset of Qu Yi Yuan. It discovered 320 (about 5% of all users) suspected ticket scalpers from the APP users of a real-world hospital.

The method first modeled event-logs into a unified process model. Then, a combined process mining and clustering approach were applied to build the hierarchical groups of event sequences. Based on the groups of event sequences, an extended DHC clustering approach was applied to discover hierarchical user group. At last, by analyzing the representative events of the center user in each group, the user profiles were created. In a particular application, various combination of user profiles could be selected and tested to hold different business requirements such as scalper detection.

In the approach, the profiles are named by human experts that actually requires domain knowledge and manual works. It is still a major challenge to automate the profile naming process. However, in a specialized application, a pre-defined profile set could be built for automating the selection. Particularly, in the future, a representative event case set could be pre-defined in the APP for profile selection.

The profiles discovered from user event logs could be interested not only in scalper detection in elderly healthcare services but also for other specialized user discovery. Applying the approach to social networks or online games are also the interesting issues for user behaviors analysis.

As future work, we plan to design a novel approach to enable a general naming process on profile selection. By combing through Linked Open Data, beyond local data environment, the profiles could be automatically discovered and selected from open-local combined data environment.

Acknowledgment

The work has been supported in part by (i) National Natural Science Foundation of China under contract number 61373030, 71171132 and 61663046; (ii) Yunnan Applied Fundamental Research Project under the Grant No. 2016FB104; (iii) Yunnan Provincial Young academic and technical leaders reserve talents under the Grant No. 2017HB005; (iv) Yunnan Provincial Science Research Project of the Department of Education under the Grant No.2018JS008.

References

[1] Alvaro Rebuge, and D. R. Ferreirab, “Business process analysis in healthcare environments: A methodology based on process mining,” Information Systems, Vol. 37, No.2, pp. 99-116, 2012.

[2] Lin, Dekang, “An Information-Theoretic Definition of Similarity,” International Conference on Machine Learning, pp.296-304, 1998.

[3] Saul B. Needleman, Christian D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” Journal of Molecular Biology, Vol. 48, no. 3, pp. 443-453, 1970.

[4] Michael Cochez, and Hao Mou,“Twister tries: approximate hierarchical agglomerative clustering for average distance in linear time,” Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 505-517, 2015

TABLE III
THE RELATED TECHNOLOGIES OF THE APPROACH

	Related Researches	Our approach
Healthcare APP	From doctor-patient communication to decision support in mobile healthcare [8-15].	Scalper detection in the mobile APP of doctor appointment.
Similarity Calculation	From string-based matching to graph-based matching [16-20]	Extending string matching with timestamps and location on event sequence matcher.
Clustering	Bottom-up clustering AHC [27-28] Up-bottom clustering DHC [23,29,30]	Extending AHC with event sequence similarity to create hierarchical clusters of events. Extending DHC with events clusters to create users clusters.
Profiling	Selecting keywords, urls, rules, attributes and preferences as profiles [31-35].	Selecting representative event of center user on each cluster as profile.

[5] Zhouyu Fu, Weiming Hu and Tieniu Tan, "Similarity based vehicle trajectory clustering and anomaly detection," IEEE International Conference on Image Processing. pp. II-602-5, 2005.

[6] Sanjoy Dasgupta, "A cost function for similarity-based hierarchical clustering," Acm Sigact Symposium ACM, pp, 118-127, 2016.

[7] Xiong, Tengke and Wang, Shengrui and Mayers, Andre and Monga, Ernest, "DHCC: Divisive Hierarchical Clustering of Categorical Data," Data Min. Knowl. Discov., Vol. 24, no. 1, pp. 103-135, 2012.

[8] Shah Jahan Miah, Najmul Hasan, Rashadul Hasan and John Gammack, "Healthcare support for underserved communities using a mobile social media platform," Information Systems, Vol. 66, pp. 1-12, 2017.

[9] J. Qi, P. Yang, G. Min, O. Amft, F. Dong and L. Xu, "Advanced Internet of Things in Personal healthcare System: A Surve", Pervasive and Mobile Computing, Vol. 41, pp. 132-149, 2017.

[10] Barjis J, Kolfschoten G, Maritz J., "A sustainable and affordable support system for rural healthcare delivery," Decision Support Systems, Vol. 56, no. 6, pp. 223-233, 2013.

[11] P. Yang, D. Stankevicius, V. Marozas, Z. Deng, E. Liu, A. Lukosevicius, F. Dong, L. Xu and G. Min, "Lifelogging Data Validation Model for Internet of Things enabled Personalized Healthcar", IEEE Transactions on Systems, Man and Cybernetics: System, Vol. 48, no. 1, pp. 50-64, 2018.

[12] J. Qi, P. Yang, M. Hanneghan, and S. Tang, "Multiple Density Maps Information Fusion for effectively assessing intensity pattern of lifelogging physical activit", Neurocomputing, Vol. 220, pp. 199-209, 2017.

[13] Praveen D, Patel A, Raghu A, et al, "SMARTHealth India: Development and Field Evaluation of a Mobile Clinical Decision Support System for Cardiovascular Diseases in Rural India," Jmir Mhealth & Uhealth, Vol. 2, no. 4, pp. e54, 2014.

[14] Fung N L S, Jones V M, Bults R G A, et al., "Guideline-based decision support for the mobile patient incorporating data streams from a body sensor network," International Conference on Wireless Mobile Communication and Healthcare, pp.312-315, 2014.

[15] Li Y P, Fang L Q, Gao S Q, et al., "Decision support system for the response to infectious disease emergencies based on WebGIS and mobile services in China,". Plos One, Vol. 8, no. 1, pp. e54842, 2013.

[16] Esposito C, Ciampi M, Pietro G D., "An event-based notification approach for the delivery of patient medical information," Information Systems, Vol. 39, no. 1, pp. 22-24, 2014.

[17] Schabetsberger T, Ammenwerth E, Andreatta S, et al., "From a paper-based transmission of discharge summaries to electronic communication in health care regions," International Journal of Medical Informatics, Vol 75, no. 3, pp. 209-215, 2006.

[18] Hillestad R, Bigelow J, Bower A, et al., "Can Electronic Medical Record Systems Transform Health Care? Potential Health Benefits, Savings, And Costs," Health Affairs, Vol. 24, no. 5, pp. 1103, 2005.

[19] D. Ritze, O. Lehmberg, and C. Bizer, "Matching HTML Tables to DBpedia," Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics, pp. 1-6, 2015.

[20] Page L., "The PageRank Citation Ranking : Bringing Order to the Web," Stanford Digital Libraries Working Paper, Vol. 9, no. 1, pp. 1-14, 1998.

[21] Jeh G, Widom J., "SimRank: a measure of structural-context similarity," International Conference on Knowledge Discovery and Data Mining, pp. 538-543, 2002.

[22] S. Melnik, H. Garcia-Molina and E. Rahm, "Similarity flooding: a versatile graph matching algorithm and its application to schema matching," Proceedings 18th International Conference on Data Engineering, pp. 117-128, 2002.

[23] C. Xie, G. Li, H. Cai, L. Jiang, and N. N. Xiong,"Dynamic weight-based individual similarity calculation for information searching in social computing," IEEE Systems Journal, vol. 11, no. 1, pp. 333-344, 2017.

[24] C. Xie, H. Cai, L. Xu, L. Jiang, and F. Bu, "Linked semantic model for information resource service towards cloud manufacturing," IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3338-3349, 2017.

[25] C. Xie, P. Yang and Y. Yang, "Open Knowledge Accessing Method in IoT-Based Hospital Information System for Medical Record Enrichment," IEEE Access, vol. 6, pp.15202-15211, 2018.

[26] Srbu C, Zehl K, Einax J W., "Fuzzy divisive hierarchical clustering of soil data using GustafsonKessel algorithm," Chemometrics and Intelligent Laboratory Systems, Vol. 86, no. 1, pp. 121-129, 2007.

[27] Gronau I, Moran S., "Optimal implementations of UPGMA and other common clustering algorithms," Information Processing Letters, Vol. 104, no. 6, pp 205-210, 2007.

[28] Kriege N, Mutzel P, Schafer T., "SAHN Clustering in Arbitrary Metric Spaces Using Heuristic Nearest Neighbor Search," International Workshop on Algorithms and Computation, pp.90-101, 2014.

[29] Mllner D., "fastcluster : Fast Hierarchical, Agglomerative Clustering Routines for R and Python," Journal of Statistical Software, Vol. 53, no. 9, pp. 1-18, 2015.

[30] Kull M, Vilo J., "Fast approximate hierarchical clustering using similarity heuristics," BioData Mining, Vol 1, no. 1, pp. 1-9, 2008.

[31] Gilpin S, Qian B, Davidson I., "Efficient hierarchical clustering of large high dimensional datasets," ACM International Conference on Information & Knowledge Management, pp. 1371-1380, 2013.

[32] Yang Y. and Jiang J., "HMM-based hybrid meta-clustering ensemble for temporal data," Knowledge-based systems, Vol. 56, pp. 299-310, 2014.

[33] Yang Y. and Jiang J., "Hybrid sampling-based clustering ensemble with global and local constitutions," IEEE Transactions on Neural Networks and Learning SystemsVol. 27, no. 5, pp. 952-965, 2016.

[34] Yang Y. and Jiang J., "Bi-weighted ensemble via HMM-based approaches for temporal data clustering," Pattern RecognitionVol. 76, pp. 391-403, 2018.

[35] Xiong T, Wang S, Mayers A, et al., "A New MCA-Based Divisive Hierarchical Algorithm for Clustering Categorical Data," IEEE International Conference on Data Mining. pp. 1058-1063, 2009.

[36] Xiong T, Wang S, Mayers A, et al., "DHCC: Divisive hierarchical clustering of categorical data," Data Mining and Knowledge Discovery, Vol. 24, no. 1, pp. 103-135, 2012.

[37] Sugiyama K, Hatano K, Yoshikawa M., "Adaptive web search based on user profile constructed without any effort from users," International Conference on World Wide Web, pp. 674-684, 2004.

[38] Al-Shamri M Y H., "User profiling approaches for demographic recommender systems," Knowledge-Based Systems, Vol. 100, pp. 175-187, 2016.

[39] Peng J, Choo K K R, Ashman H., "User profiling in intrusion detection," Journal of Network & Computer Applications, Vol. 72, pp. 14-27, 2016.

[40] Amo S D, Diallo M S, Diop C T, et al., "Contextual preference mining for user profile construction," Information Systems, Vol. 49, pp. 182-199, 2015.

[41] Servajean M, Akbarinia R, Pacitti E, et al., "Profile Diversity for Query Processing using User Recommendations," Information Systems, Vol. 48, pp. 44-63, 2014.