JISTaP http://www.jistap.org
Journal of Information Science Theory and Practice

Research Paper

J. of infosci. theory and practice 2(2): 33-47, 2014
http://dx.doi.org/10.1633/JISTaP.2014.2.2.3

Mobile User Behavior Pattern Analysis by Associated Tree in Web Service Environment

Krishna K. Mohbey *
Maulana Azad National Institute of
Technology Bhopal, India
E-mail: kmohbey@gmail.com

G. S.Thakur
Maulana Azad National Institute of
Technology Bhopal, India
E-mail: ghanshyamthakur@gmail.com

ABSTRACT

Mobile devices are the most important equipment for accessing various kinds of services. These services are accessed using wireless signals, the same used for mobile calls. Today mobile services provide a fast and excellent way to access all kinds of information via mobile phones. Mobile service providers are interested to know the access behavior pattern of the users from different locations at different timings. In this paper, we have introduced an associated tree for analyzing user behavior patterns while moving from one location to another. We have used four different parameters, namely user, location, dwell time, and services. These parameters provide stronger frequent accessing patterns by matching joins. These generated patterns are valuable for improving web services, recommending new services, and predicting useful services for individuals or groups of users. In addition, an experimental evaluation has been conducted on simulated data. Finally, performance of the proposed approach has been measured in terms of efficiency and scalability. The proposed approach produces excellent results.

Keywords: Mobile User, Mobile Location, Mobile Service, Mobile Pattern, Associated Tree

1. INTRODUCTION

Mobile devices are very popular for communication and accessing mobile services. Mobile services can be accessed anywhere via information service providers (Chen et al., 2012). By the development of wireless communication techniques and high feature mobile devices, mobile commerce has become

Open Access

Accepted date: April 29, 2014
Received date: September 17, 2013

***Corresponding Author:** Krishna K. Mohbey
PhD Research Scholar
Maulana Azad National Institute of Technology
Bhopal, India
E-mail: kmohbey@gmail.com

easier and fast. Today most of the mobile devices have high features and support for internet and web services (Kiukkonen et al., 2010). These devices are capable of sending requests for some transactions and services (Perez et al., 2010; Gao et al., 2013). They are also capable of transferring information between multiple users. Services may include selling, purchasing, online information accessing, or using social networking sites, etc. These various kinds of services are accessed by mobile devices and accessed information is stored in the transaction log files (Kiukkonen et al., 2010). Transactions mainly store user information, location information, and time information during which the user accesses any service, and finally the details of services are stored. For fast response from the system for any requested service, data mining operations can be applied to this stored information. These operations may be the mining of similar kinds of services, finding similar habits of users, and preparing groups according to services, etc. By applying such kinds of data mining operations we can restructure the web service environment and take advantage of providing recommendations and predictions for interested users. These are also helpful for business peoples for enhancing their business strategies in web environments.

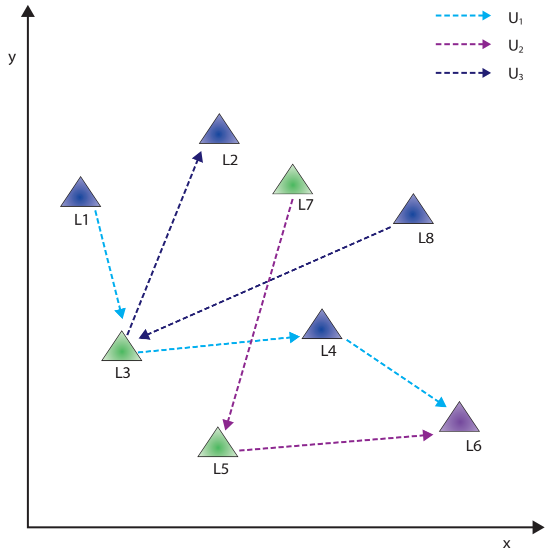

Fig. 1 Different users movement in various locations

Traditional mobile services are inadequate in handling user movement pattern analysis without the use of location, time of the service, and service request details (Lee, 2007). Over the past years some studies have employed various data mining techniques to provide interesting patterns in mobile service environments (Liu & Keselj, 2007; Pabarskaite & Raudys, 2007). Association rules are used to find unknown frequent patterns between parameters (Agrawal & Srikant, 1994). Apriori is the pioneer for mining frequent itemsets from transaction databases. FP-Growth, which is based on a pattern growth method was afterward proposed to achieve a better performance than Apriori-based methods. FP-Growth improves efficiency of frequent items because it only scans databases twice. Some studies have also focused on the uses of large databases (Cheng et al., 2004). Some recent studies are going to mine the moving path on service request log mines (Agrawal & Srikant, 1994). Researchers have also studied the problem of location tracking (Yavas et al., 2005; Tsai et al., 2007; Tseng & Lu, 2009) and resource allocation (Huang & Lin, 2008; Huang et al., 2003; Peng & Chen, 2005; Peng & Chen, 2003). Bipartite graph matching is also used to find mobile user movement behavior (Ilyas et al., 2004; Fayyazi et al., 2004; Monnot & Toulouse, 2007). Currently, different approaches are in progress to find interesting patterns of mobile users (Lee et al., 2007; Lancieri & Durand, 2006; Tseng et al., 2007; Terziyan & Vitko, 2003; Huang et al., 2003). In mobile e-commerce environments interesting user behaviors can also be predicted, based on accessed services sequences. In this paper, we analyze the behavior of the mobile user movement and find the interesting patterns. Here we have used an associated tree method for joining multiple parameters of mobile access environments. These generated patterns are helpful to business people for launching new services and for providing quick responses to existing ones. Figure 1 shows the simple scenario in which different locations are randomly visited by users. Here $l_1,l_2,l_3...l_n$ shows the different locations and $u_1,u_2,u_3...u_n$ represents different mobile users. In this proposed associated tree based pattern mining approach we have used a predefined minimum support value using those interesting patterns which are extracted from the generated candidate records. An associated tree based record represents the relationship between four predefined parameters, and its numeric value represents the degree of matching relation of a particular record.

In this paper mobile users, visited locations, dwell time, and services are used to construct the associated tree. Associated trees are used to find the frequent patterns. A Maxflow database is used to represent the relation between these four parameters and joining operations performed between them. This proposed approach is compared with the Apriori candidate generation algorithm. The remainder of this paper is organized as follows. Section 2 defines the basics of the user movement on different locations. Section 3 shows the proposed approach for mobile user behavior analysis by joining associated trees. The experiments conducted and results obtained are explained in section 4. Finally the conclusion is presented in section 5.

2. PRELIMINARIES AND PROBLEM DEFINITION

Mobile e-commerce is an emerging field for recommendation and user behavior analysis. For predicting mobile user behavior, data mining techniques are used. Various parameters are responsible for user behavior prediction. The main parameters are user, visited locations, timestamp, and services (Chen et al., 2012; Mohbey & Thakur, 2013). By these parameters, we can mine various patterns for users. According to these generated patterns, prediction and recommendations are easier. In this paper, we have generated patterns using an associated tree method and the above discussed parameters. Based on generated patterns business people are entitled to launch new services, design appropriate strategies, and so on. The following terminologies are used in this paper.

Definition 1 (Mobile User). U={u_1, u_2, u_3,...u_n}is the set of mobile users. Each mobile user represents a physical person who carries a mobile device that has the capability of receiving services from the mobile environment (Chen et al., 2012; Mohbey & Thakur, 2013).

Definition 2 (Visited Location). L={l_1,l_2,l_3,...l_n}is the set of locations at which users visit and stay for some time (Mohbey & Thakur, 2013).

Definition 3 (Timestamp). T={T_1,T_2,T_3,...T_n} is the set of time durations where mobile users spend time for using requested services. The timestamp T_n, as defined in Table 2, is assumed to have an equal period and a uniform unit. The maximum duration is considered 30 minutes in general (Chen et al., 2012).

Definition 4 (Mobile Services). S={s_1,s_2,s_3,...s_n} is the set of mobile services requested by the mobile users.

Definition 5 (Support of a Relation). The support of a relation r, such as user→location, user→time, user→service, location→time etc, is denoted as sup(r) and defined as the number of tuples that contain r in the maxflow database. For example in Table 3, the support of the relation u_1→l_1 is 3.

Definition 6 (Matching Join). If $M \subseteq U \bowtie L \bowtie T \bowtie S$. this M is a match join of U,L,T,S (Chen et al., 2012; Mohbey & Thakur, 2013).

Table 1 shows the mobile user movement database in which (l_i,Time); l_i identifies the Location and Time in HH:MM:SS format. This table is derived from Figure.1. Here 3 users visited different locations at different times.

Table 1. Database for Mobile User Movement

Time	User$_1$	User$_2$	User$_3$	...	User$_n$
T_1	(l_1,5:01:20)	(l_7,10:20:15)	(l_8,15:10:12)	...	...
T_2	(l_3,5:15:35)	(l_5,10:26:30)	(l_3,15:20:12)	...	...
T_3	(l_4,5:25:20)	(l_6,10:34:32)	(l_2,15:25:30)	...	...
T_4	(l_6,5:30:21)			...	...
.	.				
.	.				

Table 2 shows the predefined timeslots. Maximum duration taken is 30 minutes. According to this, we divide a whole day into 48 time slots of 30 minutes each. These time slots are identified as T_1,T_2,T_3...T_{48}. Each time slot indicates a particular time when a particular user accesses a service from some location.

Table 2. Predefine Time Slots for a day

Time Slot	Time Intervals	Time Slot	Time Intervals	Time Slot	Time Intervals
T_1	0:01-0:30	T_{17}	8:01-8:30	T_{33}	16:01-16:30
T_2	0:31-1:00	T_{18}	8:31-9:00	T_{34}	16:31-17:00
T_3	1:01-1:30	T_{19}	9:01-9:30	T_{35}	17:01-17:30
T_4	1:31-2:00	T_{20}	9:31-10:00	T_{36}	17:31-18:00
T_5	2:01-2:30	T_{21}	10:01-10:30	T_{37}	18:01-18:30
T_6	2:31-3:00	T_{22}	10:31-11:00	T_{38}	18:31-19:00
T_7	3:01-3:30	T_{23}	11:01-11:30	T_{39}	19:01-19:30
T_8	3:31-4:00	T_{24}	11:31-12:00	T_{40}	19:31-20:00
T_9	4:01-4:30	T_{25}	12:01-12:30	T_{41}	20:01-20:30
T_{10}	4:31-5:00	T_{26}	12:31-13:00	T_{42}	20:31-21:00
T_{11}	5:01-5:30	T_{27}	13:01-13:30	T_{43}	21:01-21:30
T_{12}	5:31-6:00	T_{28}	13:31-14:00	T_{44}	21:31-22:00
T_{13}	6:01-6:30	T_{29}	14:01-14:30	T_{45}	22:01-22:30
T_{14}	6:31-7:00	T_{30}	14:31-15:00	T_{46}	22:31-23:00
T_{15}	7:01-7:30	T_{31}	15:01-15:30	T_{47}	23:01-23:30
T_{16}	7:31-8:00	T_{32}	15:31-16:00	T_{48}	23:31-24:00

Table 3. Maxflow Database of Mobile user-location-time-service relations

Record Id	Tuples
R_1	(u_1,l_1,T_{13},s_3)
R_2	(u_3,l_2,T_{13},s_3)
R_3	(u_1,l_2,T_{38},s_1)
R_4	(u_3,l_1,T_{13},s_3)
R_5	(u_3,l_4,T_{25},s_4)
R_6	(u_1,l_1,T_{38},s_2)
R_7	(u_2,l_4,T_{25},s_3)
R_8	(u_3,l_2,T_{13},s_4)
R_9	(u_2,l_4,T_{38},s_1)
R_{10}	(u_1,l_4,T_{25},s_5)
R_{11}	(u_2,l_2,T_{25},s_3)
R_{12}	(u_3,l_4,T_{13},s_2)
R_{13}	(u_2,l_3,T_{25},s_3)
R_{14}	(u_1,l_1,T_{13},s_4)
R_{15}	(u_3,l_2,T_{25},s_3)

Table 3 shows the maxflow database in which each record represents the user-location-time-service relation. This relation is used to prepare frequent n-trees.

3. FREQUENT ASSOCIATED TREE PATTERN MINING

Associated trees are used to extract the interesting frequent patterns from the various relationships of the user-location-time-service values. Frequent tree patterns are extracted by applying loops. Then we get n-tree candidate patterns. To find the interesting frequent patterns, we firstly scan the maxflow database and generate frequent 1-tree candidate patterns. If the extracted candidate pattern is less than the predefined minimum support value then this pattern is pruned.

Now, a matching join operation is applied to generate a frequent 2-tree pattern. Again if the extracted candidate pattern support is less than the predefined minimum support then infrequent patterns are pruned. Similarly we extract the 3-tree candidate pattern and then the n+1 tree candidate frequent pattern. This process continues until the tree is empty or no more candidates are generated.

After getting each n-tree candidate frequent pattern set we prepare a associated sub tree of the relation record and count the nodes on it. Here maximum node count of a sub tree represents the strong frequent tree pattern. Algorithm 1 show the complete process of frequent associated tree pattern mining (Chen et al., 2012).

Algorithm 1. Algorithm to find frequent associated tree patterns.

```
Frequent n-Tree candidate patterns denoted as $C_n^m$, m=4.
Maximum and minimum count of nodes denoted as Max and Min respectively.
Sum of nodes denoted as SN.
For (a=0 to 4) do                // 1-tree candidate patterns
generate $C_1^4$ and $L_1$
End For
SN=0;
For (a=0 to 4) do                // 2-tree candidate patterns
        Node= generate $C_2^4$ and $L_2$ by matching join to $C_1^4$ and $L_1$
        If a=0 then
                Max =Node; Min=Node;
        Else
                If (Node > Max) then
                Max=Node;
                Else if (Node < Min) then
                Min=Node;
                End IF
        End If
SN+=Node;
End For
SN=0;
For (a=0 to 4) do                        //3-tree candidate patterns
        Node=generate $C_3^4$ and $L_3$ by matching join to $C_2^4$ and $L_2$
        If a=0 then
                Max =Node; Min=Node;
        Else
                If (Node > Max) then
                Max=Node;
                Else if (Node <Min) then
                Min=Node;
                End IF
        End If
SN+=Node;
End For
```

This whole process of finding frequent candidate patterns is illustrated in the example below.

4. ILLUSTRATIVE EXAMPLE

The complete procedure of associated tree pattern mining is shown here. For this illustration we have taken 15 records in the maxflow database which is shown in table 3. Matching join operation is performed between four parameters to generate candidate sets. These candidate sets are C_1^4, C_2^4, C_3^4, C_4^4, ..., C_n^m and frequent item sets are L_1, L_2, L_3, L_4... L_n.

4.1. Generating Candidate Set C_1^4

The proposed approach firstly generates the candidate set C_1^4 for user, location, dwell time, and services. These generated candidate sets are shown in Table 4, Table 5, Table 6, and Table 7 respectively.

Table 4. Candidate set for user

C_1^4	Count	Support
u_1	5	0.83(5/6)
u_2	4	0.67(4/6)
u_3	6	1(6/6)

Table 5. Candidate set for location

C_1^4	Count	Support
l_1	4	0.8(4/5)
l_2	5	1(5/5)
l_4	5	1(5/5)

Table 6. Candidate set for dwell time

C_1^4	Count	Support
T_{13}	6	1(6/6)
T_{25}	6	1(6/6)
T_{38}	3	0.5(3/6)

Table 7. Candidate set for service

C_1^4	Count	Support
s_1	2	0.28(2/7)
s_2	2	0.28(2/7)
s_3	7	1(7/7)
s_4	3	0.42(3/7)

4.2. Generating Candidate Set C_2^4

Now we apply a matching join operation between C_1^4 and frequent itemset L_1, then we found candidate set C_2^4 and L_2. These sets are $U_i \bowtie L_j$, $U_i \bowtie T_k$, $U_i \bowtie S_m$, $L_j \bowtie T_k$, $L_j \bowtie S_m$ and $T_k \bowtie S_m$. These matching joins are represented in Table 8 to Table 13 and corresponding frequent 2-tree candidate patterns in Figures 2 to 7 respectively.

Table 8 represents the matching join of $U_i \bowtie L_j$. It is prepared by counting the relation of $U_i \bowtie L_j$ as

$u_1 \rightarrow l_1=3$,
$u_1 \rightarrow l_2=1$,
$u_1 \rightarrow l_4=1$,
$u_2 \rightarrow l_1=0$,
$u_2 \rightarrow l_2=1$,
$u_2 \rightarrow l_4=2$,
$u_3 \rightarrow l_1=1$,
$u_3 \rightarrow l_2=3$,
$u_3 \rightarrow l_4=2$

We prune the relations which have lesser count from the predefined minimum support. In this example we have set minimum count=2; so that the relation $u_1 \rightarrow l_1$, $u_2 \rightarrow l_4$, $u_3 \rightarrow l_2$, $u_3 \rightarrow l_4$ follows the matching join condition. These relations are denoted in Table 8 and the corresponding frequent 2-tree candidate pattern is shown in Figure 2. Associated Trees prepared by various cases of the matching join $U_i \bowtie L_j$, are presented in Table 8. By joining all these cases we find the final frequent candidate tree of 13 nodes.

Table 8. Matching joins for $U_i \bowtie L_j$

C_2^4	U	L	T	S	Count	Support
$u_1 \rightarrow l_1$	1	1	T_{13}	3	3	1(3/3)
	1	1	T_{38}	2	3	1(3/3)
	1	1	T_{13}	4	3	1(3/3)
$u_2 \rightarrow l_4$	2	4	T_{25}	3	2	0.67(2/3)
	2	4	T_{38}	1	2	0.67(2/3)
$u_3 \rightarrow l_2$	3	2	T_{13}	3	3	1(3/3)
	3	2	T_{13}	4	3	1(3/3)
	3	2	T_{25}	3	3	1(3/3)
$u_3 \rightarrow l_4$	3	4	T_{25}	4	2	0.67(2/3)
	3	4	T_{13}	2	2	0.67(2/3)

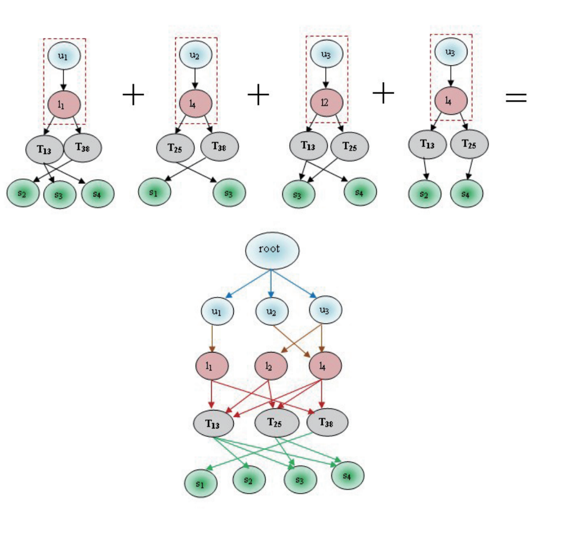

Fig. 2 Associated tree for U$_i$⋈L$_j$ of Table 8

Similarly, different C_2^4 candidate sets are shown in Table 9, Table 10, Table 11, Table 12, and Table 13. Its frequent 2-Tree is shown in Figure 3, Figure 4, Figure 5, Figure 6, and Figure 7 respectively.

Table 9. Matching joins for U$_i$⋈T$_k$

C_2^4	U	L	T	S	Count	Support
$u_1 \rightarrow T_{13}$	1	1	T_{13}	3	2	0.5(2/4)
	1	1	T_{13}	4	2	0.5(2/4)
$u_1 \rightarrow T_{38}$	1	2	T_{38}	1	2	0.5(2/4)
	1	1	T_{38}	2	2	0.5(2/4)
$u_2 \rightarrow T_{25}$	2	4	T_{25}	3	3	0.75(3/4)
	2	2	T_{25}	3	3	0.75(3/4)
	2	3	T_{25}	3	3	0.75(3/4)
$u_3 \rightarrow T_{13}$	3	2	T_{13}	3	4	1(4/4)
	3	1	T_{13}	3	4	1(4/4)
	3	2	T_{13}	4	4	1(4/4)
	3	4	T_{13}	2	4	1(4/4)
$u_3 \rightarrow T_{25}$	3	4	T_{25}	4	2	0.5(2/4)
	3	2	T_{25}	3	2	0.5(2/4)

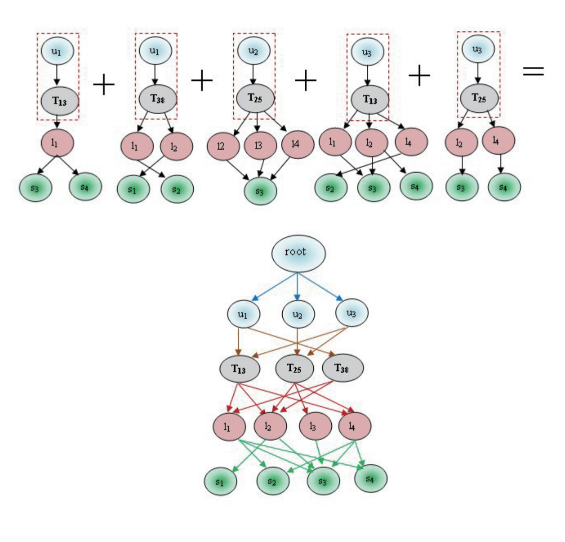

Fig. 3 Associated tree for U$_i$⋈T$_k$ of Table 9

Table 10. Matching joins for U$_i$⋈S$_m$

C_2^4	U	L	T	S	Count	Support
$u_2 \rightarrow s_3$	2	4	T_{25}	3	3	1(3/3)
	2	2	T_{25}	3	3	1(3/3)
	2	3	T_{25}	3	3	1(3/3)
$u_3 \rightarrow s_3$	3	2	T_{13}	3	3	1(3/3)
	3	1	T_{13}	3	3	1(3/3)
	3	2	T_{25}	3	3	1(3/3)
$u_3 \rightarrow s_4$	3	4	T_{25}	4	2	0.67(2/3)
	3	2	T_{13}	4	2	0.67(2/3)

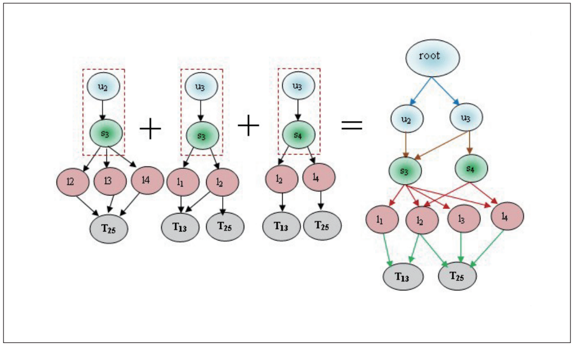

Fig. 4 Associated tree for U$_i$⋈S$_m$ of Table 10

Table 11. Matching joins for $L_j \bowtie T_k$

C_2^4	U	L	T	S	Count	Support
$l_1 \rightarrow T_{13}$	1	1	T_{13}	3	3	1(3/3)
	3	1	T_{13}	3	3	1(3/3)
	1	1	T_{13}	4	3	1(3/3)
$l_2 \rightarrow T_{13}$	3	2	T_{13}	3	2	0.67(2/3)
	3	2	T_{13}	4	2	0.67(2/3)
$l_2 \rightarrow T_{25}$	2	2	T_{25}	3	2	0.67(2/3)
	3	2	T_{25}	3	2	0.67(2/3)
$l_4 \rightarrow T_{25}$	3	4	T_{25}	4	3	1(3/3)
	2	4	T_{25}	3	3	1(3/3)
	1	4	T_{25}	5	3	1(3/3)

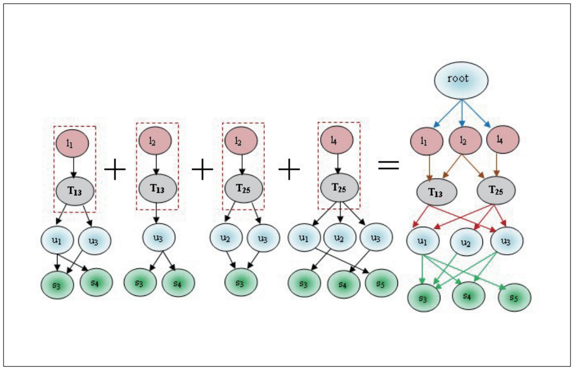

Fig. 5 Associated tree for $L_j \bowtie T_k$ of Table 11

Table 12. Matching joins for $L_j \bowtie S_m$

C_2^4	U	L	T	S	Count	Support
$l_1 \rightarrow s_3$	1	3	T_{13}	3	2	0.67(2/3)
	3	1	T_{13}	3	2	0.67(2/3)
$l_2 \rightarrow s_3$	3	2	T_{13}	3	3	1(3/3)
	2	2	T_{25}	3	3	1(3/3)
	3	2	T_{25}	3	3	1(3/3)

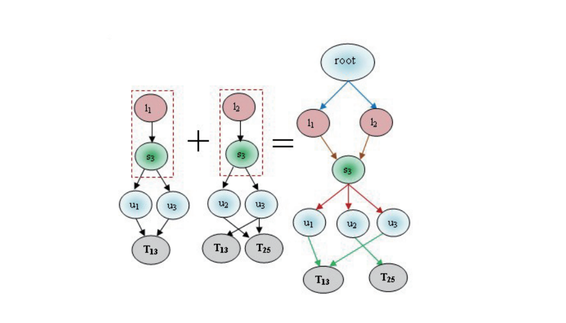

Fig. 6 Associated tree for $L_j \bowtie S_m$ of Table 12

Table 13. Matching joins for $T_k \bowtie S_n$

C_2^4	U	L	T	S	Count	Support
$T_{13} \rightarrow s_3$	1	1	T_{13}	3	3	0.75(3/4)
	3	2	T_{13}	3	3	0.75(3/4)
	3	1	T_{13}	3	3	0.75(3/4)
$T_{13} \rightarrow s_4$	3	2	T_{13}	4	2	0.5(2/4)
	1	1	T_{13}	4	2	0.5(2/4)
$T_{25} \rightarrow s_3$	2	4	T_{25}	3	4	1(4/4)
	2	2	T_{25}	3	4	1(4/4)
	2	3	T_{25}	3	4	1(4/4)
	3	2	T_{25}	3	4	1(4/4)
$T_{38} \rightarrow s_1$	1	2	T_{38}	1	2	0.5(2/4)
	2	4	T_{38}	1	2	0.5(2/4)

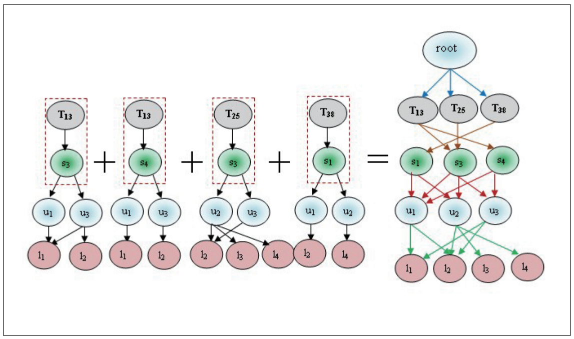

Fig. 7 Associated tree for $T_k \bowtie S_m$ of Table 13

In all candidate sets of C_2^4 we have counted nodes by associated tree as

No. of nodes in $U_i \bowtie L_j$ = 13

No. of nodes in $U_i \bowtie T_k$ = 14

No. of nodes in $U_i \bowtie S_m$ = 10

No. of nodes in $L_j \bowtie T_k$ = 11

No. of nodes in $L_j \bowtie S_m$ = 8

No. of nodes in $T_k \bowtie S_m$ = 13

By these matching joins we identified that the relation $U_i \bowtie T_k$ has the maximum node from all C_2^4 candidate sets, which is depicted in Figure 3. Thus it is clear that the relation $\mathbf{U_i \bowtie T_k}$ is stronger than the other matching relations of the candidate sets of C_2^4.

4.3. Generating Candidate Set C_3^4

Now we find the candidate sets of C_3^4 and frequent itemset L_3. These are prepared by the relation $U_i \bowtie L_j \bowtie T_k$, $U_i \bowtie L_j \bowtie S_m$, $U_i \bowtie T_k \bowtie S_m$ and $L_j \bowtie T_k \bowtie S_m$. C_3^4 candidate sets are illustrated in Tables 14 to 17 and the correspondent associated 3-Tree is shown in Figures 8 to 11 respectively.

Table 14. Matching joins for $U_i \bowtie L_j \bowtie T_k$

C_3^4	U	L	T	S	Count	Support
$u_1 \rightarrow l_1 \rightarrow T_{13}$	1	1	T_{13}	3	2	1(2/2)
	1	1	T_{13}	4	2	1(2/2)
$u_3 \rightarrow l_2 \rightarrow T_{13}$	3	2	T_{13}	3	2	1(2/2)
	3	2	T_{13}	4	2	1(2/2)

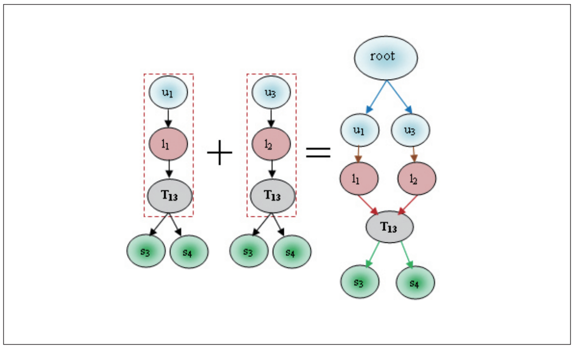

Fig. 8 Associated tree for $U_i \bowtie L_j \bowtie T_k$ of Table 14

Table 15. Matching joins for $U_i \bowtie L_j \bowtie S_m$

C_3^4	U	L	T	S	Count	Support
$u_3 \rightarrow l_2 \rightarrow s_3$	3	2	T_{13}	3	2	1(2/2)
	3	2	T_{25}	3	2	1(2/2)

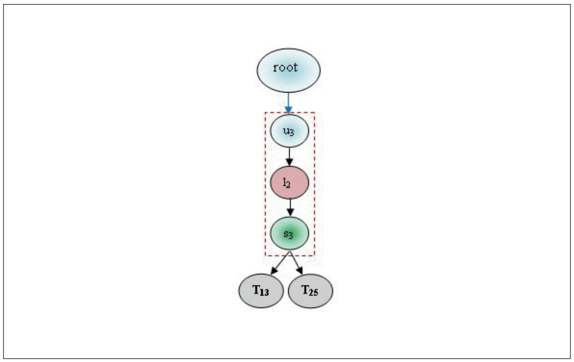

Fig. 9 Associated tree for $U_i \bowtie L_j \bowtie S_m$ of Table 15

Table 16. Matching joins for $U_i \bowtie T_k \bowtie S_m$

C_3^4	U	L	T	S	Count	Support
$u_2 \rightarrow T_{25} \rightarrow s_3$	2	4	T_{25}	3	3	1(3/3)
	2	2	T_{25}	3	3	1(3/3)
	2	3	T_{25}	3	3	1(3/3)
$u_3 \rightarrow T_{13} \rightarrow s_3$	3	2	T_{13}	3	2	0.67(2/3)
	3	1	T_{13}	3	2	0.67(2/3)

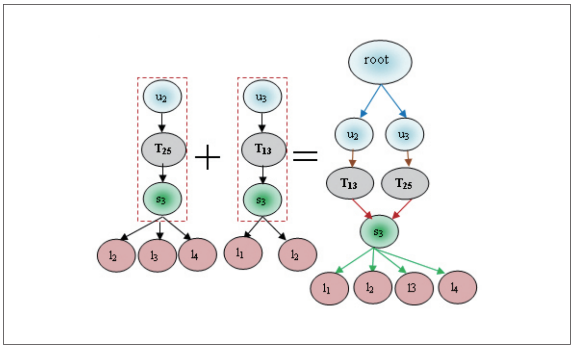

Fig. 10 Associated tree for $U_i \bowtie T_k \bowtie S_m$ of Table 16

Table 17. Matching joins for $L_j \bowtie T_k \bowtie S_m$

C_3^4	U	L	T	S	Count	Support
$l_1 \rightarrow T_{13} \rightarrow s_3$	1	1	T_{13}	3	2	1(2/2)
	3	1	T_{13}	3	2	1(2/2)
$l_2 \rightarrow T_{25} \rightarrow s_3$	2	2	T_{25}	3	2	1(2/2)
	3	2	T_{25}	3	2	1(2/2)

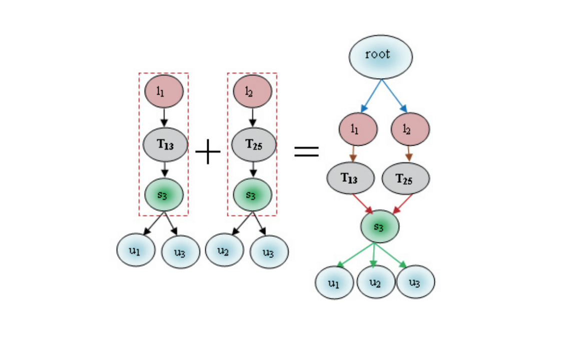

Fig. 11 Associated tree for $L_j \bowtie T_k \bowtie S_m$ of Table 17

$U_i \bowtie L_j \bowtie T_k$ are constructed by the counting of relations as

$u_1 \rightarrow l_1 \rightarrow T_{13}=2$,
$u_1 \rightarrow l_1 \rightarrow T_{38}=1$,
$u_2 \rightarrow l_4 \rightarrow T_{25}=1$,
$u_2 \rightarrow l_4 \rightarrow T_{38}=1$,
$u_3 \rightarrow l_2 \rightarrow T_{13}=2$,
$u_3 \rightarrow l_2 \rightarrow T_{25}=1$,
$u_3 \rightarrow l_4 \rightarrow T_{13}=1$,
$u_3 \rightarrow l_4 \rightarrow T_{25}=1$.

In all these counted relations we prune the relations which have less count (<2) from the predefined minimum support so that $u_1 \rightarrow l_1 \rightarrow T_{13}=2$ and $u_3 \rightarrow l_2 \rightarrow T_{13}=2$ satisfy the minimum support. These relations are denoted in Table 14 with the corresponding frequent 3-Tree in Figure 8. Similarly different C_3^4 candidate sets and associated trees are prepared. After preparing all associated sub trees the following nodes are counted:

No. of nodes in $U_i \bowtie L_j \bowtie T_k$ =7
No. of nodes in $U_i \bowtie L_j \bowtie S_m$=2
No. of nodes in $U_i \bowtie T_k \bowtie S_m$=9
No. of nodes in $L_j \bowtie T_k \bowtie S_m$=8

It is observed that the relation $\mathbf{U_i \bowtie T_k \bowtie S_m}$ has the maximum node count, thus this relation is stronger than other relations of the candidate set C_3^4.

4.4. Generating Candidate Set C_4^4

In the next step we generate the candidate set C_4^4 for relation $U_i \bowtie L_j \bowtie T_k \bowtie S_m$. It is shown in Table 18. Here we observed that the count of this relation is less than the predefined minimum support (1<2). Thus the process is terminated and the frequent item set and 4-tree candidate pattern are not prepared. Hence this relation is neglected.

Table 18. Matching joins for $U_i \bowtie L_j \bowtie T_k \bowtie S_m$

C_4^4	U	L	T	S	Count	Support
$u_1 \rightarrow l_1 \rightarrow T_{13} \rightarrow s_3$	1	1	T_{13}	3	1	1(1/1)

By the illustrated example we have found a strong associated tree among sets C_2^4 and C_3^4. Here relation $\mathbf{U_i \bowtie T_k}$ and $\mathbf{U_i \bowtie T_k \bowtie S_m}$ are the stronger relations among all the relations. These relations show that these mobile accessing patterns are the most frequent relations. By these obtained relations one can predict that, if we know the user id and time, then services can be easily identified. According to this illustration the location parameter is less important for service prediction. These mined patterns are also helpful for future prediction and recommendation of services.

5. EXPERIMENTAL EVALUATION

We evaluate the performance of the proposed algorithm in this section. The experiments were performed on a 3.0 GHz processor with 2 GB of memory and the operating system Microsoft Windows 7. The algorithms are implemented in Java. The experiment was conducted on various parameters which are shown in Table 19.

Table 19. Parameter settings

Input variable	Value	Description
D	10,000	Number of mobile access patterns
U	25	Number of users
L	20	Number of locations
T	48	Timestamps
S	15	Number of services
MS	2	Minimum support

5.1. Analysis Case 1-Different Node Counts

In this experiment, the number of mobile access patterns was set to 5,000 and 10,000. Here the different node counts are found with respect to different minimum support values. Figures 12 and 13 show that while the minimum support value increased from 0.1% to 1%, the number of nodes also decreased. Figures 12 and 13 also show that C_2^4 has the highest number of associated sub tree patterns. C_1^4 had the lowest number of associated sub tree patterns.

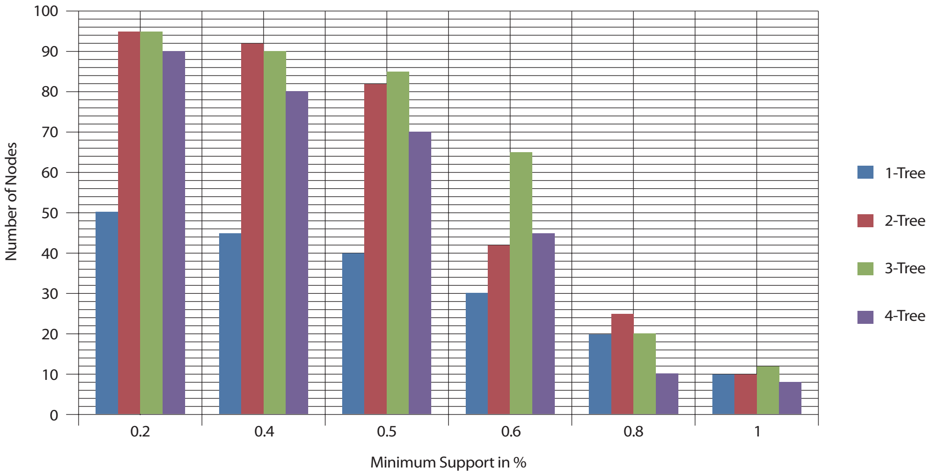

Fig. 12 No. of Nodes with different minimum support when maxflow Database=5000

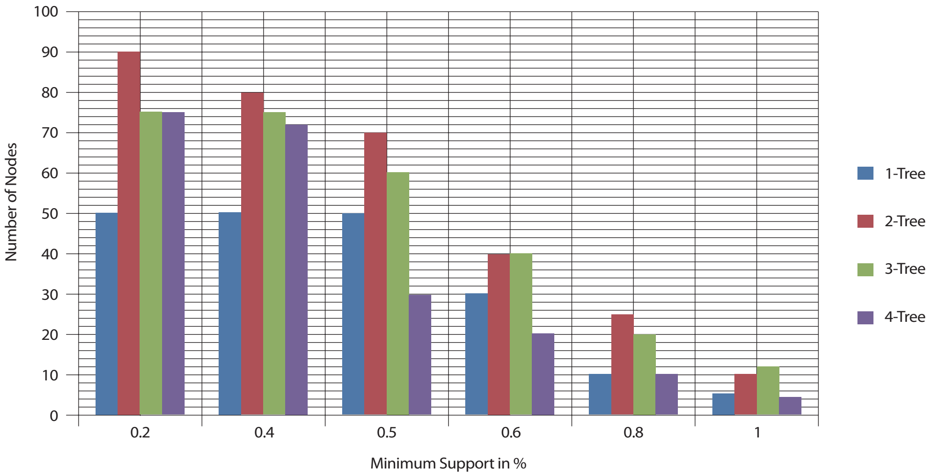

Fig. 13 No. of Nodes with different minimum support when maxflow Database=10000

5.2. Analysis Case 2-Different Item Counts

In this experiment, the number of mobile access patterns was again set to 5,000 and 10,000. Figures 14 and 15 show the performance result with respect to different minimum support values. Figures 14 and 15 show that as the minimum support value varied from 0.1% to 1% the number of items decreases.

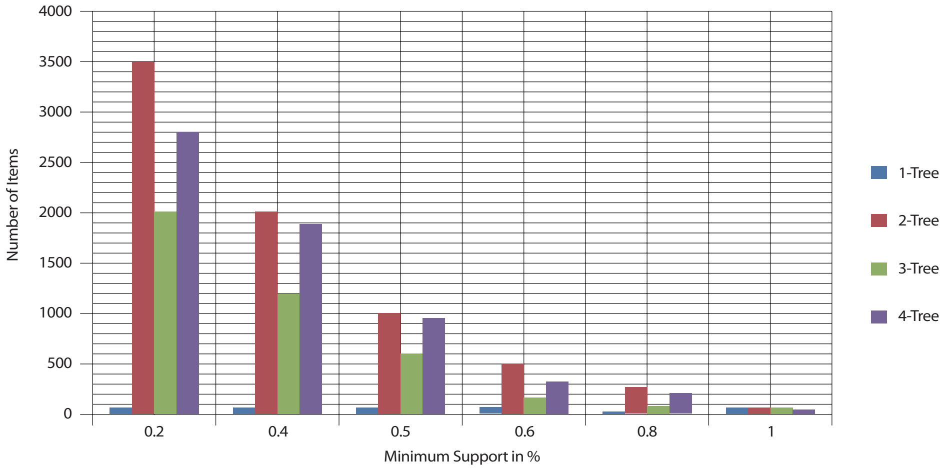

Fig. 14 No. of Items with different minimum support when maxflow Database=5000

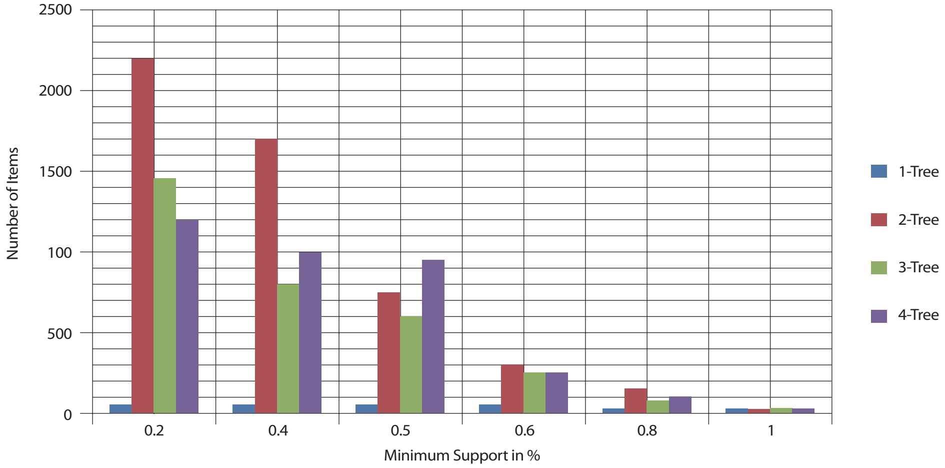

Fig. 15 No. of Items with different minimum support when maxflow Database=10000

5.3. Analysis Case 3-Different Mobile Access Patterns

In this analysis, we have compared the number of nodes in associated tree patterns and the number of mobile access patterns with respect to different minimum support values. Figures 16 to18 show these comparison results. Figure 16 shows the number of nodes and mobile access patterns with respect to min support=0.2%. Figure 16 shows that the number of nodes increased, while the number of mobile access patterns increased. The variation also shows between number of nodes and mobile access patterns while minimum support is 0.5% and 0.8% respectively. This variation is shown in Figures 17 and 18 respectively.

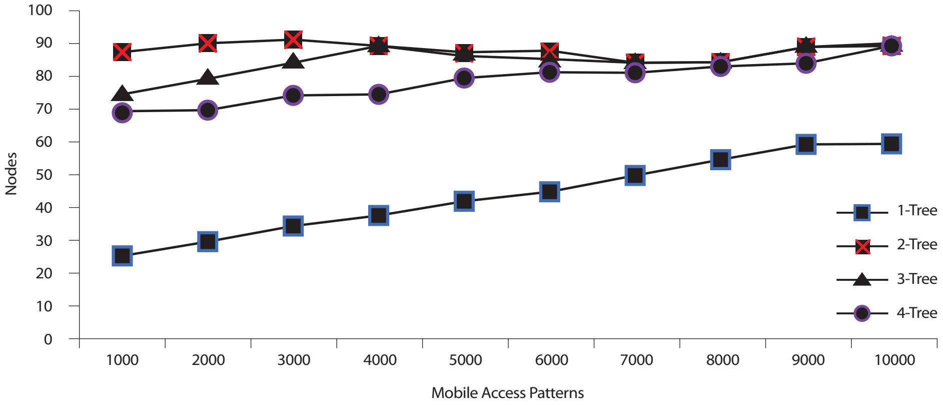

Fig. 16 Nodes and mobile access patterns with respect to 0.2% minimum support

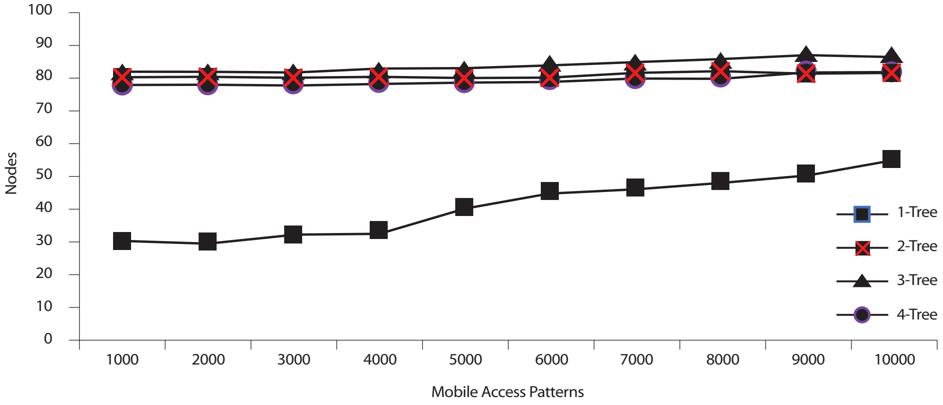

Fig. 17 Nodes and mobile access patterns with respect to 0.5% minimum support

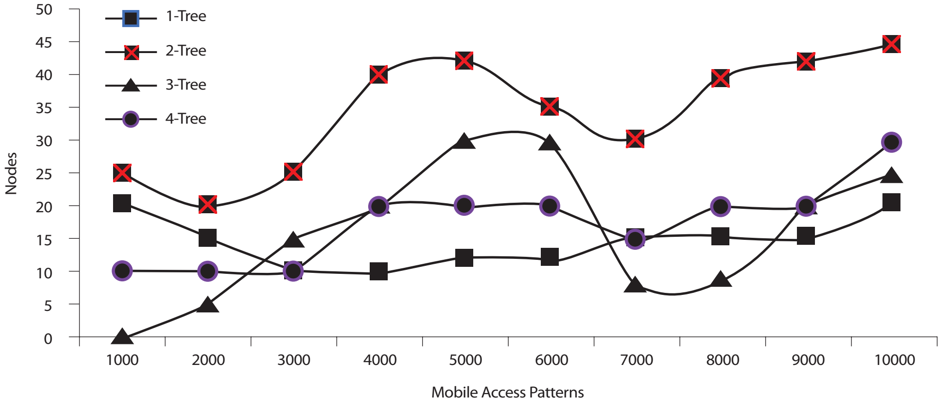

Fig. 18 Nodes and mobile access patterns with respect to 0.8% minimum support

5.4. Discussion

In this subsection experimental inferences are discussed. By the above experiments, the proposed associated tree based approach is shown to outperform the state-of-the-art approach Apriori process. The experimental results show that the proposed approach takes less time as compared to a basic association rule generation Apriori process. Execution time comparisons are shown in Figure 19. Table 20 shows the number of generated patterns with both approaches. Thus, the performance of the associated tree based approach is the best among the compared approaches. The proposed approach is also better in comparison with memory utilization.

Table 20. Number of patterns with compared approaches

Min Support (%)	Proposed approach	Apriori process
0.2	7621	12830
0.4	4310	7105
0.5	1201	3682
0.6	838	1310
0.8	430	978
1.0	123	675

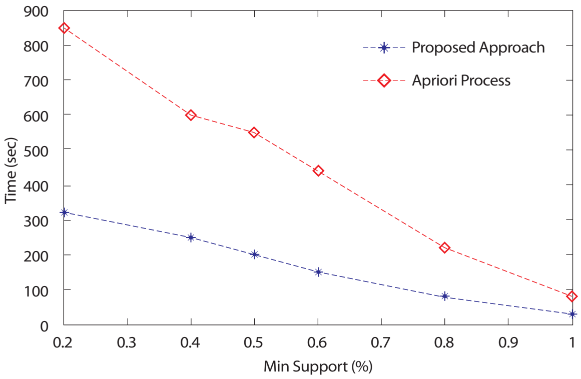

Fig. 19 Execution time in response to change in different minimum support

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an approach to generate strong frequent patterns between different mobile accessing parameters including user, location, dwell time, and web services. These patterns are generated by the matching joins and presented as associated trees. The proposed approach generates a k-tree frequent pattern in the maxflow database according to defined support values. By the experimental results under the various simulated data, the proposed approach is valuable for recommending and predicting services for different users. The proposed approach is also helpful in business and e-commerce environments where web services are important. The comparison of this approach is done with standard Apriori algorithm, in which we find that the results of the proposed approach are excellent.

For future work, we plan to explore more efficient mobile pattern generation algorithms, and design more efficient frameworks to predict mobile services. In addition, we plan to apply some specific constraints in mining processes to achieve efficient behavior of mobile users. It is also planned to find service patterns according to utility or importance bases.

ACKNOWLEDGEMENTS

The authors would like to thank editors and referees for their helpful comments.

REFERENCES

Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference of Very Large Databases*, 487-499.

Chen, T. S., Chou, Y. S. & Chen, T. C. (2012). Mining user movement behavior patterns in a mobile service environment. *IEEE Transactions on system, man, and Cybernetics Part–A Syst. Humans, 42*(1), 87-101.

Cheng, H., Yan, X. & Han, J. (2004). IncSpan: Incremental mining of sequential patterns in a large database. In *Proceedings of the 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Seattle, WA, 527-532.

Fayyazi, M., Kaeli, D. & Meleis, W. (2004). Parallel maximum weight bipartite matching algorithms for scheduling in input-queued switches. In *Proceedings of the 18th IPDPS*, Santa Fe, NM, 4-11.

Gao,H., Tang, J., Hu, X. & Liu, H. (2013). Modeling temporal effects of human mobile behavior on location-based social networks. The 22nd ACM International Conference on Information and Knowledge Management CIKM 2013, San Francisco, USA.

Huang, Y. F. & Lin, K.H. (2008). Global data allocation based on user behaviors in mobile computing environments. *Comput. Commun., 31*(10), 2420-2427.

Huang, J. L., Chen, M. S. & Peng, W.C. (2003). Exploring group mobility for replica data allocation in a mobile environment. In *Proceedings of the ACM Int. Conf. Inf. Knowl. Manage.*, New Orleans, LA, 161-168.

Ilyas, I. F., Aref, W. G., & Elmagarmid, A. K.(2004). Supporting top-k join queries in relational databases. *J. Very Large Databases, 13*(3), 207-221.

Kiukkonen, N., Blom, J., Dousse, O. & Laurila, J. K. (2010). Towards rich mobile phone datasets: Lausanne data collection campaign. ICPS 2010: The 7th International Conference on Pervasive Services, Berlin.

Lancieri, L. & Durand, N. (2006). Internet user behavior: Compared study of the access traces and application to the discovery of communities. *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, 36*(1), 208-219.

Lee, S-C., Paik, J., Ok, J., Song, I., & Kim, U-M. (2007). Efficient mining of user behaviors by temporal mobile access patterns. *Int. J. Comput. Sci. Security, 7*(2), 285-291.

Lee, W-P. (2007). Deploying personalized mobile services in an agent based environment. *Expert Syst. Appl., 32*(4), 1194-1207.

Liu, H. & Keselj, V. (2007). Combined mining of web serve logs and web contents for classifying user navigation patterns and predicting users future requests. *Data Knowl. Eng., 61*(2), 304–330.

Mohbey, K. K.& Thakur, G. S. (2013). User movement behavior analysis in mobile service environment. *British Journal of Mathematics & Computer Science, 3*(4), 822-834.

Monnot, J. & Toulouse, S. (2007). The path partition problem and related problems in bipartite graphs. *Oper. Res. Lett., 35*(5), 677-684.

Pabarskaite, Z. & Raudys, A. (2007). A process of knowledge discovery from web log data: Systematization and critical review. *J. Intell. Inf. Syst., 28*(1), 79-104.

Peng, W. C. & Chen, M. S. (2005). Shared data allocation in a mobile computing system: Exploring local and global optimization. *IEEE Trans. Parallel Distributed System, 16*(4), 374-384.

--. (2003). Developing data allocation schemes by incremental mining of user moving patterns in a mobile computing system. *IEEE Trans. Knowl. Data Eng., 15*(1), 70-85.

Perez, I. J., Cabrerizo, F. J., & Viedma, E. H. (2010). A mobile decision support system for dynamic group decision making problem. *IEEE Trans. system, management cybernetics, A, System Humans, 40*(6), 1244-1256.

Terziyan, V. & Vitko, O. (2003). Bayesian metanetworks for modelling user preferences in mobile environment. In *KI 2003: Advances in artificial intelligence.* Berlin, Germany: Springer-Verlag, 370-384.

Tsai, H. W., Chu, C. P., & Chen, T. S. (2007). Mobile object tracking in wireless sensor networks. *Comput. Commun., 30*(8), 1811-1825.

Tseng, V. S., Lu, R. H. C., & Huang, C. H. (2007). Mining temporal mobile sequential patterns in location-based service environments. In *Proceedings of the 13th ICPADS*, Hsinchu, Taiwan, 1, 1-8.

Tseng, V. S. & Lu, E. H. C. (2009). Energy-efficient real-time object tracking in multi-level sensor networks by mining and predicting movement patterns. *Journal of System & Software, 82*(4), 697-706.

Yavas, G., Katsaros, D., Ulusoy, O., & Manolopoulos, Y. (2005). A data mining approach for location prediction in mobile environments. *Data Knowl. Eng., 54*(2), 121-146.