A Novel Location Prediction Algorithm of Mobile Users For Cellular Networks

Giang Minh Duc (1), **Le Manh** (2), **Do Hong Tuan** (1)
(1) HCM City University of Technology, (2) Van Hien University
Email: ducgm.bdg@vnpt.vn, manhle@uit.edu.vn, do-hong@hcmut.edu.vn

Abstract - **Predicting the location of a mobile user is one of the important issues in mobile computing systems. Applications of the location prediction include adjusting the bandwidth of the mobile network, the location based services (LSB), smart handover, etc. However, the applications require the execution time of the User Mobility Patterns Mining (UMPMining) algorithm be instantaneous. In this paper, we propose a new algorithm named Find_UMP for mining next location of a mobile user. Our algorithm includes two phase as follows. In the first phase (Find_UMP_1), we make to reduce the complexity of the UMPMining algorithm. In the second phase (Find_UMP_2), we perform to reduce the number of transactions of the paths database. Results of our experiments show that our proposed algorithm outperforms the UMPMining algorithm in terms of the execution time.**

Keywords - Location prediction, cellular networks, Mobility prediction, Data mining.

1. INTRODUCTION

Currently, the rapid development of wireless communication technologies and the modern mobile equipments have created an environment to exchange new data. That is the mobile working environment.

Managing the path of mobile users in mobile computing environment including storage methods and location updating are served by the system. An important issue in the field is location prediction of the mobile users.

The movement of a mobile user from a present cell to another cell will be recorded in a database that is called the Visitor Location Register (VLR). Then, the data will be transferred to the Home Location Registry (HLR) [12 - 14] that are located at the Mobile Switching System (MSC).

Location Prediction of the mobile users is used to increase the efficiency of the mobile network [11, 17, 18]. Therefore, the system can allocate resources efficiently to cells instead of allocating spreading to all the cells in network.

In addition, it also has a number of applications as follows:

- The service providers can calculate in an optimal way when they design structure and bandwidth of mobile network [1].
- The telecommunications service providers can reduce the number of unnecessary handover in hierarchical macro/femto-cell networks [16].
- Location Based Services (LBS) [15, 19].
- Etc…

The sequential mining algorithm was applied to the mobility prediction for cellular networks [8, 10, 15, 16] known as User Mobility prediction Algorithm (UMPMining Algorithm).

Our algorithm has the running time reduced by 75.18% compared to the UMPMining algorithm. Find_UMP_1 algorithm reduces the complexity of the algorithm by changing the data structure (Section 3). Find_UMP_2 algorithm performs to reduce the number of transactions (Section 4).

This paper is organized as follows. In section 2, we present the related works. In section 3 and section 4, we present our algorithms. The results of the experiments are presented in section 5. In section 6, mobility rules are extracted from the user mobility patterns. In section 7, the user mobility prediction is implemented by these rules. The conclusion is given Section 8.

2. RELATED WORKS

The problem of sequential pattern mining has been mentioned in [1 - 3]. In addition, the sequential patterns mining are also applied to predict the accessing of users on the Web [4], [5]. *Web prefetching* is defined as delivering of future requirements of the users based on the previous requests. Ignorant Prediction Method [6] does not pay attention to valuable information in the user moving history. In order to predict mobility of user, this method assigns a number of neighboring cells and performs to choose randomly m cells adjacent to the current cell.

Mobility Prediction Method based on Transition Matrix (TM) [7] predicted location according to the ability could occur transition "cell-to-cell" of a mobile user is calculated by the previous move and then recorded in a matrix. Relying on this basis, the allocation of resources is done in k cells most likely in the neighboring cell. The parameter k is a parameter defined by the user.

UMPMining algorithm [8, 9, 10, 15, 16] predicts the location of mobile users using data mining techniques.

The algorithm in [10] is also the same as [8], [15], but the paths storage file of mobile users is stored in the grid node placed at different locations. Data grid provides a geographic distributing database for computational Grid and executes by an algorithm called KMPM (Knowledge Grid Based Mobility Pattern Mining). If the number of nodes increases, the computation time of the algorithm decreases.

Following are two algorithms that we propose:

3. FIND_UMP_1 ALGORITHM

In order to reduce the complexity of the UMPMining algorithm in [8, 15, 16], we mapped the path database of mobile users (the database D) to path matrix M_{dd} (Definition 5). Steps are as follows:

Suppose that UAPs have form as follows:

$C = \{c_1, c_2 ... c_n\}$. Each c_k denotes the ID number of the cell k_{th} in coverage area.

For example, we have the coverage map simulated as follows:

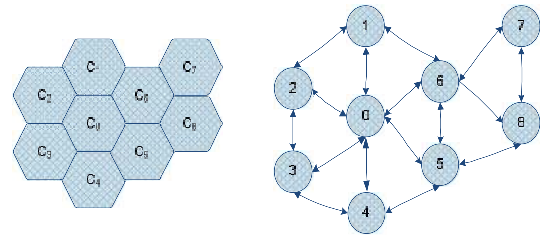

Figure 1: Simulate of cellular network and graph G

Table 1: Paths of mobile users

UAP ID	UAPs
1	{5,6,0,4}
2	{3,4,5,0}
3	{1,2,3,4,0,5}
4	{3,2,0}

Data table 1 as follows:

We call the regular moving patterns of users are UMPs. By mining the UMPs, we will get the mobility rules.

G is called a directed graph corresponds with cells in the mobile coverage area. Each cell of the G is a peak as Figure 1. If there are two cells that called A, B neighboring each other (bordering common) in the mobile coverage area, they will have a path direct and weightlessness from A to B and backwards.

Definition 1: Data Mining Context

Give O is a non-empty limited set of transactions (UAP ID) and I is a non-empty limited set of cells, R is a two subject relation between O and I such that $o \in O$ and $i \in I$, $(o, i) \in R \Leftrightarrow$ transaction o contains cell i_{th}. The data mining context is the triple (O, I, R).

Definition 2: Data Mining Context Matrix

Give a *mobile user's paths* table includes two properties that are UAP_ID (code of a transaction) and UAP (path of a mobile user through cells of the mobile coverage map). Call O is a set of transactions. I is a set of cells and R is a two subject relation between O and I, $R \subseteq O \times I$, where $(o, i) \in R$ if and only if transaction o is contained cell i_{th}.

Definition 3: Galois Connection

Give a data mining context (O, I, R), consider two functions ρ and λ, they are defined as follows: $\rho\ P(I) \rightarrow P(O)$ and $\lambda\ P(O) \rightarrow P(I)$:

Give $S \subseteq I$, $\rho(S) = \{o \in O \mid \forall i \in S, (o, i) \in \mathbf{R}\}$

Give $X \subseteq O$, $\lambda(X) = \{i \in I \mid \forall o \in X, (o,i) \in \mathbf{R}\}$

Where P(X) is a set of subsets of X. A pair of function (ρ,λ) is defined in such that way called Galois Connection.

ρ(S) value denotes a set of transactions that have common all cells in S. λ(X) value denotes a set of cells that have in all transactions of X.

Property 1: a pair of function (ρ,λ) have properties as follows

1.1 Where $S_1, S2 \in P(I)$, $S_1 \subseteq S_2 \Rightarrow \rho(S_2) \subseteq \rho(S_1)$

1.2 Where $X_1, X2 \in P(O)$, $X1 \subseteq X_2 \Rightarrow \lambda(X_2) \subseteq \lambda(X_1)$

1.3 $S \subseteq \lambda(\rho(S))$ and $X \subseteq \rho(\lambda(X))$

1.4 $\lambda(\rho(\lambda(X))) = \lambda(X)$ and $\rho(\lambda(\rho(S))) = \rho(S)$

Definition 4: frequent set

Give a data mining context (O, I, R), and $S \subset I$, the frequent level of S is defined as ratio between the number of transactions and all of transactions of O. the frequent of S is called SP(S) and it is computed as follows:

$$SP(S) = |\rho(S)| / |O|$$

Where $|\,.\,|$ is the length of set.

Give $S \subset I$ and minsupp is a minimum frequent threshold, S is a frequent set of minsupp threshold if and only if $SP(S) \geq minsupp$.

FS (O, I, R, minsupp): is a set of frequent subsets by the minsupp threshold or FS (O, I, R, minsupp) = $\{S \in P(I) \mid SP(S) \geq minsupp\}$

Clause 1:

Give $T \notin FS(O, I, R, minsupp)$, if $T \subseteq S$, $S \notin FS(O, I, R, minsupp)$

Demonstration: due to $T \subseteq S$, according to property (1.1) of the Galois Connection of a pair of function (ρ, λ), we have $\rho(S) \subseteq \rho(T)$, therefore $SP(S) \leq SP(T) < minsupp \rightarrow S \notin FS(O,I,R,minsupp)$.

Definition 5: M_{dd} mobility Matrix

The M_{dd} Mobility Matrix is similar to the Binary Matrix as definition 2, but it is added as follows: each M $[O_m, i_n]$ is a location of a mobile user traveling in mobile network (table 2).

Column i_n: code of a cell in mobile network.

Row o_m: UAP ID of a mobile user.

We exchange data from table 1 to table 2 as follows:

Table 2: Mobility matrix of mobile users

	i_0	i_1	i_2	i_3	i_4	i_5	i_6	i_7
o_1	3	0	0	0	4	1	2	0
o_2	4	0	0	1	2	3	0	0
o_3	5	1	2	3	4	6	0	0
o_4	3	0	2	1	0	0	0	0

For example, in Table 2, the second mobile user (UAP ID = 2) moves between the cells as follows:

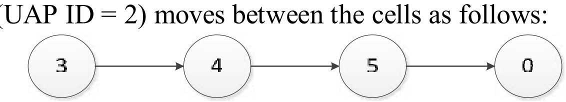

Figure 2: Mobility of the second mobile user

Therefore, the path o_2 is performed the follows:

$o_2 = (4, 0, 0, 1, 2, 3, 0, 0)$

Besides, we must consider to the mobile coverage of graph G (figure 1).

The following is a new algorithm to find UMPs from the mobility matrix M_{dd}:

Find_UMP_1 algorithm

Input: minsupp, M_{dd}, G
Output: L

```
L = ∅   // initially the frequent patterns set is empty
L1 ← Find_L1 //generalize L1 from Find_L1 function
For (k=2; Lk-1 ≠ ∅; k++) do
    Lk ← Find_Lk(Lk-1) //generalize Lk from Lk-1
    L = L ∪ Lk
endfor
Return L
```

At line 2, we have Find_L_1() function as follows:

Find_L_1 algorithm: find L_1

Input: (O, I, R), minsupp, M_{dd}, G
Output: L_1.

1. $L_1 = \varnothing$
2. For each ($i \in I$ and $j \in$ field of M_{dd}) //i: cell //ID and it is also a column of M_{dd}
3. $S=\{s \mid s \in M_{dd}$ and $s_{ij} \neq 0\}$
4. For each $s \in S$
5. s.count = s.count + 1
6. Endfor
7. Endfor
8. $L = \{s \mid s \in C_1, s.count \geq minsupp\}$
9. $L_1 = L_1 \cup L$
10. Return L_1

At line 4 of Find_UMP_1 algorithm, we have a function finds L_k from L_{k-1} as follows:

Find_L_k (L_{k-1}) algorithm: create L_k from L_{k-1}

Input: L_{k-1}, G, M_{dd}
Output: L_k

1. $L_k = \varnothing$
2. For (each $X \in L_{k-1}$) do
3. For (each $Y \in L_{k-1}$ and $X \neq Y$) do
4. $S = X \cup Y$
5. $S = \{s_1,s_2,\ldots,s_{k-1},s_k\}$ //s_k: a set of cells has link with s_{k-1} of G
6. if ($|S| = k$ and $SP(S) \geq minsupp$) then
7. $L_k = L_k \cup \{S\}$
8. Endif
9. Endfor
10. Endfor
11. Return L_k

At line 6 of Find_L_k(), we have a function finds support of S_k as follows:

Find_support(S_k) algorithm

Input: S_k, M_{dd}
Output: $SP(S_k)$

1. For each $o \in M_{dd}$ do //scan all M_{dd}
2. Find location $(s_1,s_2,\ldots,s_k) \in S_k$ of o
3. Find S_k.count
4. End for
5. Return $SP(S_k)$

The complexity of the Find_support(S_k) algorithm:

- For the loop at line 1: the complexity is O(m), where m = |O|: total records of M_{dd}
- As such, the complexity of the algorithm is O(m).

The complexity of the Find_support algorithm reduces n times (reduces one loop) compared to the UMPMining algorithm.

As running the Find_UMP_1 algorithm with a real data set as follows: 56198 records and 351 BTSs (as in table 3), the execution time of this algorithm is 410 seconds compared to the UMPMining algorithm is 548 seconds (reduce 138 seconds, corresponding 25.18% runtime).

4. FIND_UMP_2 ALGORITHM

The Find_UMP_2 algorithm is similar to the Find_UMP_1 algorithm but they differ from the function to find the support, as follows:

Reducing the number of transactions:

From the clause 1, we have:

Give $T \notin FS(O,I,R,minsupp)$, if $T \subseteq S$, $S \notin FS(O,I,R,minsupp)$.

Find_Supp_2 (S_k) algorithm: find $SP(S_k)$

Input: M_{dd}, S_k, minsupp, G
Output: $SP(S_k)$

1. Dem_dong = 1
2. **If** $|S_k| = 2$ then //scan all rows of M_{dd} ($o_n \in M_{dd}$)
3. **For** ($i = 1$; $i \leq |O|$; i++) do
4. If ($S_k \in O$ and $(s_1,s_2,\ldots,s_k)$ have order in O) then
5. Mang_tam ← save variable i
6. Mang_tam ← count the number of rows
7. End if
8. **Endfor**
9. **Else** //$|S_k| > 2$

```
10.   S_{k-1} ← S_k
11.   |O_R| ← Mang_luu  //|O_R|: the rows contain S_{k-1}
12.   For (i = 1; i ≤ |O_R|; i++)     //|O_R| < |O|
13.    If (S_k ∈ O_R and (s_1,s_2,…,s_k) have order in O_R)
        then
14.        Mang_tam ← save variable i
15.        Mang_tam ← count the number of rows
16.     Endif
17.    Endfor
18. Endif
19. SP(S_k) ← Find support
20. If SP(S_k) > minsupp then
21.    Mang_luu ← Mang_tam
22. Endif
23. Return SP(S_k)
```

Comment:

- Case $|S_k| = 1$: the method, which calculates the support of the Find_support(S_k) and the Find_support_2(S_k), is the same. Therefore, the execution time of them is equal (shown Table 3).
- Case $|S_k| = 2$: the method, which calculates the support of the Find_support_2 (S_k), is added line 6 ÷ 7, 21 ÷ 22 with the following meanings:

  If $SP(S_k) \geq$ minsupp, we save these rows in an array(Mang_luu) including a number of the suited rows and S_k.supp.

  Our purpose is to decrease a number of loop time as finding S_{k+1}. From clause 1, we show: if $S_k \notin FS(O,I,R,minsupp)$ and $S_k \subseteq S_{k+1}$, then $S_{k+1} \notin FS(O,I,R,minsupp)$. For example, if $S_k = \{3, 2\}$ and $SP(S_k) = 1 \leq$ minsupp = 1.33, $S_{k+1} = \{3, 2, 1\}$ $\leq$ minsupp.

  This algorithm is implemented with a actually database as follows:

  Input data UAPs have the number of paths as follows: 56 198 (all rows of matrix M_{dd}: $|O|$ = 56 198).

  A number of BTSs are 351 (a number of fields of matrix M_{dd}: $|I|$ = 351).)
- Case $|S_k| \geq 2$:

  At line 11 ÷ 12: we get from mang_luu the previous information as the number of rows contained S_{k-1}, is O_R (to reduce the number of loop).

  At line 13 ÷ 18: instead of scanning all database, we just implement loop the O_R times.

5. EXPERIMENTAL RESULTS

5.1. Compare the execution time of the algorithms: UMPMining, Find_UMP_1 and Find_UMP_2

When $|S_k|$=2, such as $S_k = \{1,7\}$:

- $SP(S_k) = 16.5 \geq$ minsupp = 2.5;
- The rows of the M_{dd} matrix contains S_k ($S_k \subseteq O_R$ và $O_R \in M_{dd}$):

O_R = {2456, 3789, 3791, 4233, 4234, 4241, 4606, 11748, 15194, 22594, 25349, 29813, 34873, 38544, 43806, 46084, 47730}

- The number of rows satisfies $S_k \subseteq O_R$: 17.

It is the loop when $S_k = \{1, 7\}$ (line 11, 12 of Find_Supp_2), that is $|O_R|$ = 17. Thus, we find that instead of scanning all database ($|O|$ = 56198), we just scan 17 records (reduced 56181 records).

The actual statistic:

- The number of the loop when running Find_UMP_1(): 192.028.566 times
- The number of the loop when running Find_UMP_2(): 110.672 times.
- Thus, the number of reducing is 191.917.894 times (Figure 3).

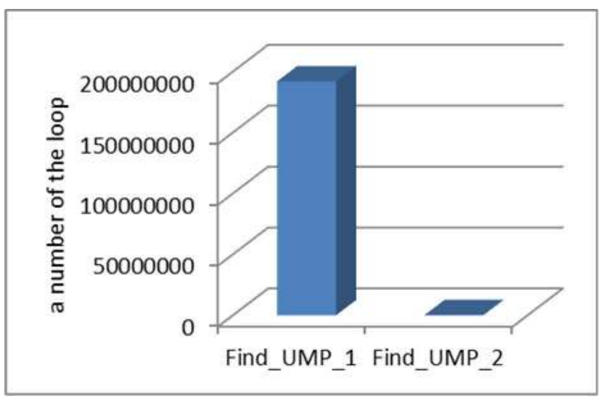

Figure 3: Compare the number of the loop of two algorithms

Thus, we find that the running time of the algorithm Find_UMP_2 () reduced as follows: only 136 seconds (table 3), reduced 66, 82% (compared to the Find_UMP_1 algorithm) and reduced 75.18% (compared to the UMPMining algorithm).

We have results as follows:

Table 3: Compare results of three algorithms

C_n	UMPMining		Find_UMP_1		Find_UMP_2	
	quantity C_n	Run time	quantity C_n	Run time	quantity C_n	Run time
C_1	351	32	351	1	351	1
C_2	1488	167	1488	129	1488	129
C_3	3340	341	3340	274	3340	5
C_4	79	8	79	6	79	1
Total	**5258**	**548**	**5258**	**410**	**5258**	**136**

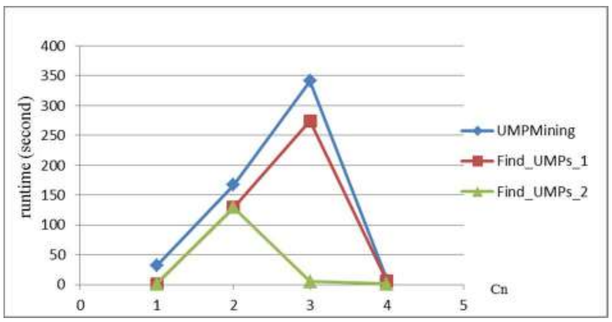

Figure 4: Compare the running time results of three algorithms

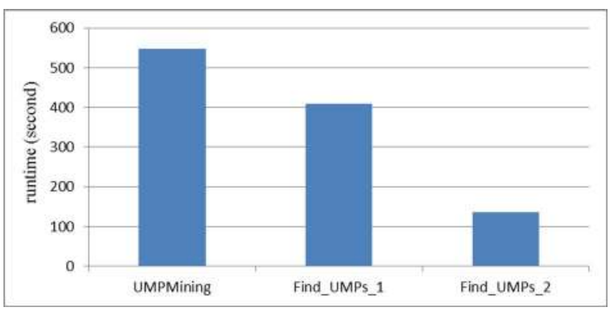

Figure 5: Compare the running time total of three algorithms

5.2. The accuracy of the prediction

Training and testing datasets are an important part of evaluating data mining models [20].

In our experiments, training dataset and testing dataset(*) are the actual database of mobile users. The database is transformed from the User ID to the integer n (n = 1, 2, 3...) and they cannot be decoded to protect customer's information. Training dataset and testing dataset include sets given in Table 4.

Table 4. Training and Testing datasets

Name	The number of transactions of mobile users
Training dataset	56198
Testing dataset	7207

The testing dataset is UAPs; it is used to evaluate the accuracy of the users' mobility prediction.

Testing dataset contains 7207 transactions of mobile users.

The number of BTS: 351.

We use two following parameters [8] for the evaluation of our algorithms.

- Recall: the number of correctly predicted cells/ the total number of requests.
- Precision: the number of correctly predicted cells/ the total number of predictions made.

• Changing of the recall values according to the minsupp values:

In Figure 6, if the minsupp value increases, then the recall value decreases. The reason is the increasing minsupp value will make the number of prediction rules reduces. Therefore, the number of correctly predictions is reduced.

(*) Appendix in References

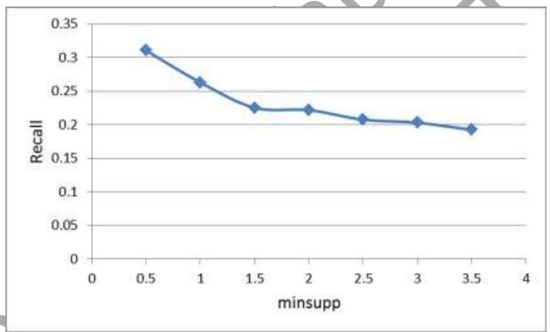

Figure 6: Changes of recall according to minsupp of the dataset

- The precision of the prediction rules when changing the minconf value:

When changing the minimum confidence value (minconf), the precision value changes as Figure 7.

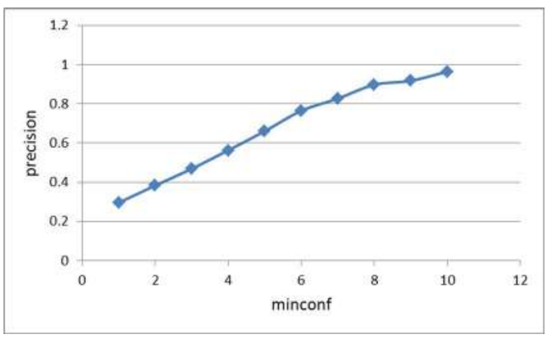

Figure 7: Precision of the prediction rules

In Figure 7, if the minconf value increases, then the precision value increases. The reason is at high minconf values, only the rules that have high confidence values are used for mobility prediction of mobile users.

As the above result, the number of prediction rules reduces but the quality gets higher with the increasing minconf value.

6. FINDING THE MOBILITY RULES

The results from the data mining phase (UAPs → UMPs); we are the mobility patterns of mobile users (UMPs). In this section, we will find the mobility rules from UMPs.

For example: we have a form UMP is (3, 4, 5). The mobility rules as follows:

$(3) \rightarrow (4, 5)$

$(3, 4) \rightarrow (5)$

Supposed we have a UMP L = $\{i_1, i_2 \ldots i_k\}$, where $k > 1$. All mobility rules originated from the pattern as follows:

$\{i_1\} \text{ -> } \{i_2,\ldots, i_k\}$

$\{i_1, i_2\} \text{ -> } \{i_3,\ldots, i_k\}$

…

$\{ i_1, i_2\ldots i_{k-1}\} \text{ -> } \{ i_k \}$

Give the mobility rule R: $(i_1, i_2\ldots i_{i-1}) \rightarrow (i_i, i_{i+1}\ldots i_k)$, the confidence value is calculated as follows:

$$Confidence(R) = \frac{(i_1, i_2, \ldots, i_k).count}{(i_1, i_2, \ldots, i_{i-1}).count} \times 100$$

By using UMPs, all mobility rules are created and the confidence value is calculated. Rules have confidence $\geq conf_{min}$ will be selected.

7. PREDICTION THE MOBILITY OF MOBILE USERS

In this section, the next path of the user is predicted. Supposing the mobile user has his path until now is P = $(c_1, c_2 \ldots c_i)$. The algorithm finds out the rules that the arrow left contained in P and the final cell $a_j = c_i$. Save these rules into the Luat_Dubao set, we save the first cell of the right side of Luat_Dubao with confidence values in an array called Mang_Luat. The Mang_Luat is ordered in descending according to the value of confidence with the aim of selecting the most reliable value.

For example, the current user is in cell No. 4 position. The algorithm will find out the rules: (3, 4) → (0) and (4) → (5). The first right cell will be saved along with the confidence and support values in the Mang_Luat array.

8. CONCLUSION

In this paper, we presented a novel algorithm for next location prediction of mobile users. This is the stage applied the data mining techniques to find the frequent moving patterns of mobile users. Based on the result of implementing the algorithm with real dataset, we show that the execution time of the Find_UMP algorithm (Find_UMP_1 and Find_UMP_2) reduced greatly when compared to UMPMining algorithm.

In the future work, we will study to reduce more than the run time of the algorithm when new data is added. The benefits of applying these algorithms are the system can run online in real-time to monitor the flow of mobile networks.

REFERENCES

[1] A. Aljadhai, T. Znaiti, "Predictive mobility support for QoS provisioning in mobile wireless environments", *IEEE J. Select. Area Commun*. (2001) 1915-1930.

[2] H. Cao, N. Mamoulis, and D. W. Cheung. Mining frequent spatio-temporal sequential patterns. In *Proceedings of the International Conference on Data Mining (ICDM'05)*, 2005, pp. 82–89.

[3] J. Yang and M. Hu. TrajPattern: mining sequential patterns from imprecise trajectories of mobile objects. In *Proceedings of the International Conference on Extending Database Technology (EDBT'06)*, LNCS 3896, 2006, pp. 664–681.

[4] A. Nanopoulos, D. Katsaros, Y. Manolopoulos, "Effective prediction of web user accesses: a data mining approach", *in: Proceedings of the WebKDD Workshop* (WebKDD'01), 2001.

[5] A. Nanopoulos, D. Katsaros, Y. Manolopoulos, "A data mining algorithm for generalized web prefetching", *IEEE Trans. Knowl. Data Eng*. 15 (5) (2003) 1155–1169.

[6] A. Bhattacharya, S. K. Das, LeZi-Update: "an information-theoretic approach to track mobile users in PCS networks", *ACM Wireless Networks 8* (2-3) (2002) 121-135.

[7] S. rajagopal, R.B. Srinivasan, R. B. Narayan, X.B.C. Petit, "GPS-based predictive resource allocation in cellural networks", *in: Proceedings of the IEEE International Conference on Networks* (IEEE ICON020, 2002, pp. 229-234.

[8] Gokhan Yavas, dimitrios Katsaros. Ozgur Ulssoy and Yannis manolopoulos. "A data mining approach for location prediction in mobile environments". *Data and Knowledge Engineering*, 54, 121-146, 2005.

[9] Cristian Aflori and Mitica Craus. "Grid implementation of Apriori algorithm. Advances in engineering software". *Volume 38, Issue 5*, 295-300, 2007.

[10] U. Sakthi, R.S. Bhuvaneswaran: "Mobility Prediction of Mobile Users in Mobile Environment Using Knowledge Grid", Journal of Computer Science (vol.9, No.1): 303-309, 2009.

[11] ETSI/GSM. Technical reports list. http://webapp.etsi.org/key/key.asp? full list=y.

[12] ETSI/GSM. Home location register/visitor location register – report 11.31–32.

[13] Alex Cabanes (IBM Systems & Technology Group): IBM BladeCenter - Home Location Register (HLR). June 2007.

[14] HRL Look Up – Service Manual (www.routomessaging.com).

[15] M. Abo-Zahhad, Sabah M. Ahmed, M. Mourad, " Services and Applications Based on Mobile User's Location Detection and Prediction", *Int. J. Communications*, *Network and System Sciences*, 2013, 6, 167-175.

[16] Byungjin Jeong, Seungjae Shin, Ingook Jang, Nak Woon Sung, and Hyunsoo Yoon, "A Smart Handover Decision Algorithm Using Location Prediction for Hierarchical Macro/Femto-Cell Networks " *in Vehicular Conference* (VTC Fall), 2011 IEEE 74th, SanFrancisco, CA, Sept 2011, pp. 1-5.

[17] Monreale, A., Pinelli, F., Trasarti, R, Giannotti, F.: "Wherenext: a location predictor on trajectory pattern mining". *In: 15th ACM SIGKDD Conference on Knoledge Discovery and Data Mining* (KDD'09) (2009)

[18] J. J. Ying, W. Lee, T. Weng, and V. S. Tseng. "Semantic trajectory mining for location prediction". *In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43. ACM, 2011.

[19] Lu, E. H.-C.; Tseng, V. S.; and Yu, P. S. 2011. "Mining cluster-based temporal mobile sequential patterns in location-based service environments". *IEEE Trans. Knowl. Data Eng*. 23(6):914–927.

[20] http://msdn.microsoft.com/en-us/library/bb895173 .aspx (Training and Testing Data Sets – MSDN – Microsoft).

Appendix: The dataset used for experiments

✓ The dataset used for our experiments was chosen at a province of Vietnam with 351 BTSs (Base Transceiver Stations) and 1,179,034 UAPs (User Actual Paths). After the data normalization, the UAPs dataset is 56 198 records.

✓ To mine data from the HLR (Home Location Register), we perform the data normalization through four steps as follows:

Step 1: exchange data from a text file into a structured data file (database)

- Data from the Home Location Register (HLR) with the following text:

0,452028500564855,84945859880,353191034572720,20111001,082142,43,0985477139,MTC,848,5924,,,,717,17522,,,,0,4A40EB0B11,0,2011-10-01 08:21:42

0,452022020361130,84915749135,356919030975830,20111001,082123,62,01234348491,MOC,,,769,1666,,712,12442,,,,1,35414F04A7,0,2011-10-01 08:21:23

…

Step 2: Linking cell_ID

In the log file retrieved from the HLR, each BTS has a cell_ID, which links into a BTS management file of the province.

Step 3: Extracting some necessary fields of this dataset for data mining.

Step 4: Filter out records that have only one cell (mobile users do not move).

After four steps for the data normalization, we get the following database:

STT	UAP_ID	UAP
16	841224	15,128
17	841224	249,113
18	841225	136,29
19	841226	63,56
20	841226	21,223
21	841232	8,29
22	841232	319,309
23	841232	212,234,212
24	841232	124,142,124,204,221
25	841232	29,30
26	841232	170,178,170
27	841232	12,43,12
28	841232	54,63,54,63,54,63,79,63
29	841232	19,1,19,1
30	841232	282,263
31	841232	66,110
32	841232	297,301
33	841232	126,130,126,130,126,130,126,130,126
34	841232	13,11
35	841232	313,305,343,295,313,37,305,335,305,313,307
36	841232	4,13,4
37	841232	312,114
38	841232	306,309,306

AUTHORS' BIOGRAPHIES

Giang Minh Duc received the B.E degree (1994) in Electrical Engineering from Ho Chi Minh City University of Technology and B.S degree (1999) in Information Technology from The University of Science - Vietnam National University, Ho Chi Minh City. He received the M.S degree (2006) in information technology (computer science) from University of Information Technology - Vietnam National University, Ho Chi Minh City. Currently he is doing Ph.D. at Ho Chi Minh City University of Technology - Vietnam National University, Ho Chi Minh City.

His research interest includes telecommunications Engineering, Information Technology (knowledge base, data mining)

Le Manh is a lecturer in University of Information Technology - Vietnam National University, Ho Chi Minh City. Van Hien University

He received a B.S degree (1971) in Computer Engineering from Hanoi University of Science and Technology and Ph.D. in Mathematical Foundation of Computers and Computing Systems from Soviet Russia Academy of Science (1982)

His research interest includes management and development of IT applications in mobile environments and computer networks.

Do Hong Tuan is PhD, Senior Lecturer, Head of Department of Electrical and Electronics, Ho Chi Minh City University of Technology - Vietnam National University, Ho Chi Minh City.

His research interest includes Smart Antennas, Mobile and Wireless Communications, Linear and Nonlinear Microwave Circuits, Digital Image Processing.