ISSN 2047-3338

An Effective Approach for Mobility Prediction in Wireless Network based on Temporal Weighted Mobility Rule

Thuy Van T. Duong and Dinh Que Tran

Abstract— **In the wireless network, whenever a mobile node moves from one cell to another - called *handover* or *handoff*, the call needs to be handed off to the new base station, and then network resources must be reallocated. Many mobility prediction schemes are proposed to perform resource reservations in advance so as to reduce the handover latency. Such approaches make use of knowledge patterns of location being mined from the mobility history of users to describe and predict the movement of mobile users. In addition to the location characteristic, the time-of-day also plays a crucial role in modeling the movement and it has attracted several research interests recently. In this paper, we investigate simultaneously spatial and temporal attributes of data and apply a spatiotemporal data mining technique to discover frequent mobility patterns for predicting the next location of a mobile node. Our approach is to mine frequent mobility patterns with time and then to make use of them to construct temporal weighted mobility rules. This paper extends a mobility prediction algorithm by finding the best matched rules which are temporally closest to the query time. Our experimental results show that using the temporal attribute is necessary for improving the prediction accuracy.**

Index Terms— **Mobility Prediction, Pattern Mining, Rule and Spatiotemporality**

I. INTRODUCTION

IN next generation wireless networks, one of the most serious challenges is how to achieve continuous connection during mobile user movement among cells which is allowed due to handover procedure. Whenever a mobile node moves from one cell to another, the call needs to be handed off to the new base station, and network resources must be reallocated. During a handover, although the connection is still alive, a mobile node can not send or receive any packets thus the packet loss may occur as well [1]. The high delay of handover procedure is a limitation to achieve a seamless handover with the meaning that the user is unaware of such status. Hence, the handover latency needs to be reduced as much as possible for a seamless mobility. In handover procedure, resource allocation takes a lot of time and becomes the main factor contributing to the handover latency. One of the most effective approach to reduce the delay in resource allocation is to predict the next location of the mobile node. Mobility prediction detects the identity of the future cell for resources reservation prior to the actual handover [2] and it has attracted several research interests [2], [3], [4], [5].

The fact that behavior of mobile users often follows a sequence of regular movement locations has the important meaning. For example, in a university, lecturers often move to classrooms, laboratories, library, etc whereas departmental staffs often travel around the administrative offices. Therefore, it is possible to predict the next location of a mobile user based on his own movement history. According to [6], [7], [8], [9], [10], data mining has been a useful approach to mobility prediction based on mobility history. In fact, the mobility history hide useful knowledge patterns that describe typical behavior of mobile nodes. These knowledge patterns, which are represented in the form of mobility rules, can be used to describe and to predict the movement of mobile nodes. However, the current studies on mobility prediction are not satisfactory because of lack of investigating simultaneously spatial and temporal attributes of data. In the context of wireless network, the spatial attributes of a mobile node are changing over time, therefore time constraints between locations need to be considered in predicting the mobility.

In this paper, we investigate the time-of-day factor and the importance of the time when a mobile user moves to the location daily. For instance, the movement of lecturers depends on the schedule of classes. It means that he will move to the classrooms at the times depending on his own teaching schedule. Our approach is to consider simultaneously time and space factors based on spatio-temporal data mining techniques. Moreover, in order to overcome the problem of long processing time and computational expensiveness in data mining techniques, we propose that just the prediction phase is performed in an online manner whereas the frequent mobility patterns and mobility rules are discovered periodically in offline. Researching efficient data mining techniques is not the purpose of this work, so we apply the most popular Apriori technique. However, any data mining techniques could be used with our proposed approach.

Thuy Van T. Duong is with the Department of Information Technology and Applied Mathematics, University of Ton Duc Thang, Ho Chi Minh City, Vietnam (email: vanduongthuy@yahoo.com, dttvan@itam.tdt.edu.vn).

Dinh Que Tran is with the Faculty of Information Technology, Post and Telecommunication Institute of Technology, Km10, NguyenTrai, Hadong, Hanoi, Vietnam (corresponding author, phone: +84904066883; e-mail: tdque@yahoo.com, tdque@ptithcm.edu.vn).

As another result of this research, the aim of mining frequent mobility patterns is generating frequent mobility rules which are near to user's current movement for an accurate prediction. It means that the rules which are extracted from the recent mobility patterns should be more important than the ones extracted from the older patterns. To reach this goal, each generated rule is assigned a weighted value based on temporal attribute. Our experimental results show that using the temporal attribute is necessary for improving the prediction accuracy.

The rest of this paper is constructed as follows. Section 2 describes the mobility model in wireless network. Section 3 formalizes mobility in wireless network with combining space and time. Section 4 proposes a mining algorithm to discover spatiotemporal mobility patterns. Section 5 is mobility prediction based on temporal weighted rules. Section 6 presents our experiments and evaluation on many different parameters. Finally, Section 7 draws concluding remarks and further work.

II. MODELING MOBILITY IN WIRELESS NETWORK

A. Typical Wireless Network Architecture

This subsection is an overview of a typical wireless network architecture [4], [11]. In wireless network, the radio coverage region is partitioned into geographical areas, called location areas (LAs). Every LA consists of a group of cells (1, 2 or more cells). Each cell is served by a base station (BS) that assigns radio frequencies, or channels, to each mobile node (MN) within the cell. The BSs regularly broadcast the identifiers of their cells and thus the MNs will know in which cell they are now through listening to the broadcast channel. All neighboring cells in a LA are managed by a base station controller (BSC) which is used to control all the BSs in the LA for performing their jobs and a mobile switch service center (MSC) which maintains a visitor location register (VLR). The VLR records information about the MNs currently in the cells served by the MSC. Specially, the wireless network architecture also has a database which records the movements of each MN from its current cell to another cell. This database is called home location register (HLR). Therefore, in this paper it is possible to get the movement history of a MN from the log data stored in HLR. Fig. 1 shows a typical wireless network architecture.

In this paper, it is assumed that the radio coverage region is represented by a hexagonal shaped network (shown as Figure 2). Each hexagon is a cell which is served by a BS in the communication space. The mobile nodes can travel around the coverage region. In order to illustrate the mobility model of mobile nodes in wireless network, we use an unweighted directed graph $G = (V, E)$, where the vertex-set V is the set of cells in the coverage region and the edge-set E represent the adjacence between pairs of cells. That means, if two cells, say A and B, are neighboring cells in the coverage region then G has a directed and unweighted edge from A to B and also from

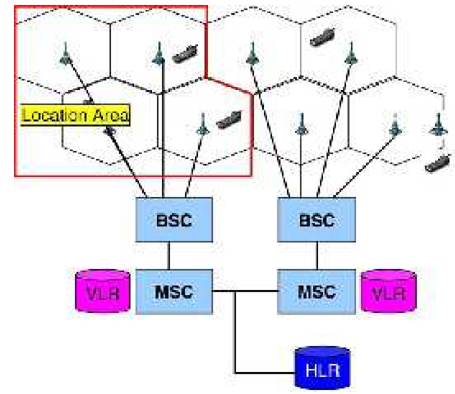


Fig. 1. A typical wireless network architecture

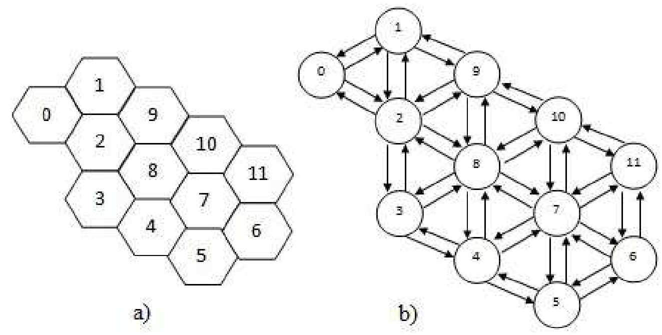


Fig. 2. An example coverage region (a) and the corresponding graph G (b)

B to A. These bidirected edges illustrate the fact that a mobile node may move from A to B or B to A directly and further may travel around the coverage region corresponding graph G which is called bidirected graph. The example network shown in Figure 2 can be modeled by the vertex-set $V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ and the edge-set $E = \{(0, 1), (0, 2), (1, 0), (1, 2), (1, 9), (2, 0), (2, 1), (2, 3), (2, 8), (2, 9), ..., (11, 6), (11, 7), (11, 10)\}$.

B. Representing Mobility Profiles with Timestamps

Behavior of mobile users can be characterized in many different ways [8]. In this work, two characteristics which are used to define mobility behaviors are location and time-of-day. The following is some discussion of the motivation for using these characteristics.

The location factor indicates that the movement of mobile users often follow a sequence of locations every day. For example, in a campus network, lecturers often move to classrooms, laboratories, library, etc whereas departmental staffs often travel around the administrative offices. Therefore, it is possible to predict the next location of a mobile user based on his own location history.

The time-of-day factor identifies the importance of the time when the mobile user moves to the location. The mobility behaviors changes as a function of time. For example, the movement of lecturers depends on the schedule of classes. That means, he will move to the classrooms at the times depending on his own teaching schedule. In this work, we

TABLE I
PREDEFINED TIMESTAMPS

Timestamps	Time Interval	Timestamps	Time Interval
t_1	0:00 - 2:14	t_7	13:00 - 15:44
t_2	2:15 - 4:29	t_8	15:45 - 17:59
t3	4:30 - 6:44	t_9	18:00 - 20:14
t4	6:45 - 8:59	t_{10}	20:15 - 22:29
t_5	9:00 - 11:14	t_{11}	22:30 - 23:59
t_6	11:15 - 13:29		

analyze the personal mobility patterns of lecturers in a campus wireless network. The movement of each lecturer depends on his own teaching schedule. Therefore, it is possible to set the time interval every three teaching periods. A teaching periods is 45 minutes so the time interval is 135 minutes. According to our work, the predefined timestamps are illustrated in Table I.

TABLE II
A LOG FILE OF NODE MOBILITY HISTORY

Time	Location	Time	Location
2011/7/19/6/40	0	2011/7/21/16/1	4
2011/7/19/6/50	2	2011/7/21/18/0	7
2011/7/19/11/25	8	2011/7/21/19/3	6
2011/7/19/13/30	3	2011/7/22/6/30	9
2011/7/19/15/45	4	2011/7/22/7/00	2
2011/7/19/18/00	7	2011/7/22/11/3	8
2011/7/20/6/45	2	2011/7/22/16/4	4
2011/7/20/9/00	3	2011/7/22/18/0	5
2011/7/20/11/30	8	2011/7/22/18/3	7
2011/7/20/15/50	4	2011/7/23/6/55	2
2011/7/20/18/00	5	2011/7/23/11/4	8
2011/7/20/19/10	7	2011/7/23/15/4	4
2011/7/21/6/58	2	2011/7/23/18/1	5
2011/7/21/11/30	8	2011/7/23/18/4	7
2011/7/21/13/30	3	2011/7/23/19/2	6

Table II is an example log file of mobility history which records the movements of a lecturer arround the coverage region graph G (See Fig. 2).

III. FORMALIZING MOBILITY OF WIRELESS NETWORK

In this subsection, we give the formal definitions to model mobility patterns and rules. As discussed in Section II, the mobile node can travel around the coverage region corresponding graph G. Let c be the ID number of the cell to which the mobile node connected at the timestamp t, we define a point as follows:

Definition 1. *Let C and T be two sets of ID cells and timestamps, respectively. The ordered pairs* $p = (c, t)$, *in which* $c \in C$ *and* $t \in T$, *is called a point. Denote P to be the set of all points* $P = C \times T = \{(c, t) \mid c \in C \text{ and } t \in T\}$.

Two point $p_i = (c_i, t_i)$ and $p_j = (c_j, t_j)$ are said to be equivalent if and only if $c_i = c_j$ and $t_i = t_j$. Point $p_i = (c_i, t_i)$ is defined to be earlier than point $p_j = (c_j, t_j)$ if and only if $t_i < t_j$, and it is denoted as $(c_i, t_i) < (c_j, t_j)$ or $p_i < p_j$.

Definition 2. *The trajectory of the mobile node is defined as a finite sequence of points* $<p_1, p_2, \ldots, p_k>$ *in* $C \times T$ *space, where point* $p_j = (c_j, t_j)$ *for* $1 \leq j \leq k$. *A sequence composed of k elements is denoted as a k-pattern.*

Note that the value of each timestamp t_j is not unique in a trajectory, i.e. t_j may be equal to t_i if they are timestamps of two consecutive points of a trajectory. For example $<(c_1, t_1), (c_2, t_2), (c_3, t_2), (c_4, t_4)>$ is a trajectory. The ascending order of points of trajectory is sorted by using t as the key.

Definition 3. *A mobility pattern* $B = <b_1, b_2, \ldots, b_m>$ *is a sub-pattern of another mobility pattern* $A = <a_1, a_2, \ldots, a_n>$, *where* a_i *and* b_j *are points, written as* $B \subset A$, *if and only if there exists integers* $1 \leq i_1 < \ldots < i_m \leq n$ *such that* $b_k = a_{ik}$, *for all k, where* $1 \leq k \leq m$. *And then, A is called the super-pattern of B.*

For example, given $A = <(c_4, t_2), (c_5, t_3), (c_6, t_4), (c_8, t_5)>$ and $B = <(c_5, t_3), (c_8, t_5)>$. Then B is a sub-pattern of A and conversely, A is super-pattern of B.

Definition 4. ([12]) *Let* $D = \{S_1, S_2, \ldots, S_N\}$ *be a transaction database which contains N sequential mobility patterns. The support of pattern S is defined as*

$$support(S) = \frac{\left|\{S_i \mid S \subset S_i \text{ and } 1 \leq i \leq N\}\right|}{N} \quad (1)$$

Definition 5. *Given a minimum support threshold,* $supp_{min}$, *a sequential mobility pattern S is defined as a frequent mobility pattern if and only if S has support value satisfying:*

$$support(S) \geq supp_{min}$$

Definition 6. *A mobility rule has the form* $R: A \rightarrow B$, *in which A and B are two frequent mobility patterns and* $A \cap B = \emptyset$. *Then, A and B are called the head and the tail of the rule, respectively.*

Since the mobility rule $R: A \rightarrow B$ is generated from the frequent mobility pattern $A \cup B$, the support of the rule is the support of the pattern $A \cup B$, that is:

$$support(R) = support(A \cup B)$$

Definition 7. ([4]) *Given a rule* $R: A \rightarrow B$, *its confidence value is determined by the formula:*

$$confidence(R) = \frac{support(A \cup B)}{support(A)} \times 100 \quad (2)$$

Problem Statement: Given a bidirected graph G, a log file H of a node mobility history, a maximal time gap, $gap_{\max}$, a set T of predefined timestamps, a minimum support threshold $supp_{\min}$, and a minimum confidence threshold $conf_{\min}$. The problem of mobility prediction in wireless networks based on spatiotemporal data mining is composed of two phases:

(i) The first phase is to discover all frequent mobility patterns in transactional database satisfying $supp_{\min}$ from the generated transactional database of a log file of a node mobility history.

(ii) The second phase is to generate all mobility rules using the frequent mobility patterns which are mined in the previous phase. The mobility rules which have confidence values higher than the predefined threshold value $conf_{\min}$

are used to predict the next cells of the given mobile node in the last part of this study.

IV. DISCOVERING SPATIO-TEMPORAL MOBILITY PATTERNS

A. Transactional Database

For discovering frequent mobility patterns from the mobility history, we extract all mobility sequences using log file of node mobility history. A set of ordered mobility sequences is called a transactional database. In order to overcome the problem of data missing, loss of connection [5] [13], we introduce the time constraints between locations of a node to make significant mobility sequences (valid transactions). Time constraints restrict the time gap between two locations in a transaction. That means, only when the time between two locations stays within the maximal time gap, called $gap_{\max}$, can a mobility sequence be produced. Let t_j and t_{j+1} denote the occurrence time point of consecutive movements then $t_{j+1} - t_j \leq gap_{\max}$.

For example, if the $gap_{\max}$ is 8 hours, then the transactional database extracted from the log file of mobility history in Table II is presented in Table III. In this example, the time gap between the sixth and the seventh location exceeds the $gap_{\max}$

TABLE III
AN EXAMPLE TRANSACTIONAL DATABASE *D*

Transaction ID	Mobility patterns
1	<(0, t₃), (2, t₄), (8, t₆), (3, t₇), (4, t₈), (7, t₉)>
2	<(2, t₄), (3, t₅), (8, t₆), (4, t₈), (5, t₉), (7, t₉)>
3	<(2, t₄), (8, t₆), (3, t₇), (4, t₈), (7, t₉), (6, t₉)>
4	<(9, t₃), (2, t₄), (8, t₆), (4, t₈), (5, t₉), (7, t₉)>
5	<(2, t₄), (8, t₆), (4, t₇), (5, t₉), (7, t₉), (6, t₉)>

of 8 hours, thus a new transaction is generated. The result shows that using temporal attribute to generate transactional database allows a more flexible handling of the transactions.

Notice that the time values of the temporal attribute in transactional database are transformed into corresponding Predefined Timestamps using the Time Intervals (See Table I).

The following subsections present an algorithm, which is the modified version of the Apriori technique [9] [14], to discover all frequent mobility patterns from the transactional database *D*. For presentation convenience, we assume that the bidirected graph *G* has *N* vertices and the maximal timestamp is *T*.

B. Discovering Frequent Mobility 1-patterns

First, all frequent mobility 1-patterns are extracted from the database *D* by the following procedure. To discover frequent 1-patterns, for each cell ID c_i for $1 \leq i \leq N$, we scan the transactional database *D* to find all points (c_i, t_j) for $1 \leq j \leq T$. Each point (c_i, t_j) is a 1-pattern. Then their support values are calculated by using (1). All 1-patterns which have support values higher than a predefined minimum support threshold ($supp_{\min}$) are selected and called frequent mobility 1-patterns as Definition 5.

TABLE IV
MOBILITY 1-PATTERNS AND FREQUENT MOBILITY 1-PATTERNS

C_1		L_1	
Candidates 1-patterns	Support	Frequent 1-patterns	Support
$\langle(0, t_3)\rangle$	1	$\langle(2, t_4)\rangle$	5
$\langle(2, t_4)\rangle$	5	$\langle(3, t_7)\rangle$	2
$\langle(3, t_5)\rangle$	1	$\langle(4, t_8)\rangle$	4
$\langle(3, t_7)\rangle$	2	$\langle(5, t_9)\rangle$	3
$\langle(4, t_7)\rangle$	1	$\langle(6, t_9)\rangle$	2
$\langle(4, t_8)\rangle$	4	$\langle(7, t_9)\rangle$	5
$\langle(5, t_9)\rangle$	3	$\langle(8, t_6)\rangle$	5
$\langle(6, t_9)\rangle$	2		
$\langle(7, t_9)\rangle$	5		
$\langle(8, t_6)\rangle$	5		
$\langle(9, t_3)\rangle$	1		

Let L_1 be a set of frequent mobility 1-patterns

$$L_1 = \left\{\left\langle\left(c_i, t_j\right)\right\rangle \mid support\left(\left\langle\left(c_i, t_j\right)\right\rangle\right) \geq supp_{min} \text{ for } 1 \leq i \leq N \text{ and } 1 \leq j \leq T\right\}$$

Example 1: Let $T = 11$ (See predefined timestamps in Table I), $supp_{\min} = 2$, transactional database is in Table III and bidirected graph is in Figure 2, then L_1 is in Table IV. Note that the support values of 1-patterns $\langle(0, t_3)\rangle$, $\langle(3, t_5)\rangle$, $\langle(4, t_7)\rangle$ and $\langle(9, t_3)\rangle$ are less than $supp_{\min}$ so they are discarded and called *outliers* (See C_1 in Table IV).

C. Discovering Frequent Mobility k-patterns for k ≥ 2

For $k \geq 2$, candidate k-patterns are discovered as follows. Given a frequent (k-1)-pattern $F=\langle(c_1, t_1), (c_2, t_2), \ldots, (c_{k-1}, t_{k-1})\rangle$. Let $V(c_{k-1})$ be a set of cells which are neighbors of c_{k-1} in G

$$V(c_{k-1})=\{v \mid v \text{ is a neighbor of } c_{k-1}\}$$

TABLE V
MOBILITY 2-PATTERNS AND FREQUENT MOBILITY 2-PATTERNS

C_2		L_2	
Candidates 2-patterns	Support	Frequent 2-patterns	Support
$\langle(2, t_4), (3, t_7)\rangle$	2	$\langle(2, t_4), (3, t_7)\rangle$	2
$\langle(2, t_4), (8, t_6)\rangle$	5	$\langle(2, t_4), (8, t_6)\rangle$	5
$\langle(3, t_7), (4, t_8)\rangle$	2	$\langle(3, t_7), (4, t_8)\rangle$	2
$\langle(4, t_8), (5, t_9)\rangle$	2	$\langle(4, t_8), (5, t_9)\rangle$	2
$\langle(4, t_8), (7, t_9)\rangle$	4	$\langle(4, t_8), (7, t_9)\rangle$	4
$\langle(5, t_9), (6, t_9)\rangle$	1	$\langle(5, t_9), (7, t_9)\rangle$	3
$\langle(5, t_9), (7, t_9)\rangle$	3	$\langle(7, t_9), (6, t_9)\rangle$	2
$\langle(6, t_9), (5, t_9)\rangle$	0	$\langle(8, t_6), (3, t_7)\rangle$	2
$\langle(6, t_9), (7, t_9)\rangle$	0	$\langle(8, t_6), (4, t_8)\rangle$	4
$\langle(7, t_9), (5, t_9)\rangle$	0	$\langle(8, t_6), (7, t_9)\rangle$	5
$\langle(7, t_9), (6, t_9)\rangle$	2		
$\langle(8, t_6), (3, t_7)\rangle$	2		
$\langle(8, t_6), (4, t_8)\rangle$	4		
$\langle(8, t_6), (7, t_9)\rangle$	5		

For each $v \in V(c_{k-1})$, generating all points (v, t_k) satisfing $1 \leq t_k \leq T$, and then $\langle(v, t_k)\rangle$ is a frequent 1-pattern and $t_k \geq t_{k-1}$. Let

$$P(c_{k-1})=\{p=(v, t_k) \mid \langle(v, t_k)\rangle \in L_1 \text{ and } t_k \geq t_{k-1}\}$$

For each $p \in P(c_{k-1})$, a candidate k-pattern C is generated by attaching $p=(v, t_k)$ to the end of F:

$$C=<(c_1, t_1), (c_2, t_2), \ldots, (c_{k-1}, t_{k-1}), (v, t_k)>$$

Then adding C to the set of candidates k-pattern:

$$C_k=C_k \cup C$$

This procedure is repeated for all the frequent (k-1)-patterns in L_{k-1}. Then, all candidates k-pattern which have support values higher than $supp_{min}$ are selected:

$$L_k=\{C| C \in C_k \text{ and } support\ (C) \geq supp_{min}\}$$

Algorithm 1 Frequent Mobility Patterns Discovery Algorithm

Input: a transactional database, D
minimum support threshold, $supp_{min}$
coverage region directed graph, G

Output: a set of frequent mobility patterns, L

```
// Let Ck is a set of candidates k-patterns
// Let Lk is a set of frequent mobility k-patterns
L1  ← a set of frequent mobility 1-patterns
k = 1
repeat
   Ck+1 ← CandidateGeneration(Lk)
   for all mobility pattern F ∈ D do
     C ← {c|c ∈ Ck+1 and c ⊂ F}
     for all c ∈ C do
       c.count = c.count + 1;
     end for
   end for
   Lk+1 ← {c|c ∈ Ck+1 and c.count ≥ suppmin}
   L = L ∪ Lk+1
   k = k+1
until Lk = ∅
return L
```

Algorithm 2 Candidates Generation Algorithm

Input: a set of frequent mobility k-patterns, L_k
coverage region directed graph, G

Output: a set of candidates (k+1)-patterns, C_{k+1}

```
for all frequent mobility k-pattern
    Pk=<(c1, t1), (c2, t2), …, (ck, tk)> ∈ Lk do
  V(ck) ← {v|v is a neighbor of ck}
  for all vertex v ∈ V(ck) do
    P(ck) ← {p=(v, tk+1)| <(v, tk+1)> ∈ L1
                                   and tk+1≥tk}
    for all p=(v, tk+1) ∈ P(ck) do
      C=<(c1, t1), (c2, t2), …, (ck, tk), (v, tk+1)>
      Ck+1= Ck+1 ∪ C
    end for
  end for
end for
return Ck+1
```

***Example 2 (continued)**:* With k = 2, for frequent mobility 1-pattern <(8, t_6)>, we have:

$V(8) = \{2, 3, 4, 7, 9, 10\}$

$P(8) = \{(3, t_7), (4, t_8), (7, t_9)\}$

$C_2 = \{<(8, t_6), (3, t_7)>, <(8, t_6), (4, t_8)>, <(8, t_6), (7, t_9)>\}$

All candidates 2-patterns discovered from L_1 in Table IV are represented by C_2 in Table V. L_2 in Table V shows all frequent mobility 2-patterns.

The pseudo-code for discovering all the frequent mobility patterns is presented in Algorithm 1.

V. MOBILITY PREDICTION BASED ON WEIGHTED RULES

A. Mining Temporal Weighted Mobility Rules

Let S be a frequent mobility pattern, all the possible mobility rules which can be generated from S are: $A \rightarrow (S\text{-}A)$ for all $A \subset S$ and $A \neq \varnothing$. For example, the frequent mobility pattern P= <(c_1, t_1), (c_2, t_2), …, (c_k, t_k)>, where $k > 1$, has all possible rules:

Algorithm 3 Mobility Rules Generation Algorithm

Input: a set of frequent mobility patterns, L
minimum confidence threshold, $conf_{min}$

Output: a set of frequent mobility rules, *Rules*

```
Rules ← ∅
for all frequent mobility pattern k-pattern
   Pk=<(c1, t1), (c2, t2), …, (ck, tk)> ∈ Lk, k≥2 do
  Pl ← Pk
  repeat
    //A ← dropping the last point of Pl i.e. A is a (l-1) subpattern of Pl
     A ← <(c1, t1), (c2, t2), …, (cl-1, tl-1)>
     conf = support(Pk)/support(A)
     if conf ≥ confmin then
       //R ← (A → Pk-A)
       R ← {<(c1, t1), (c2, t2), …, (cl-1, tl-1)>
                  →<(cl, tl), …, (ck, tk)>}
       R.w = (RuleDate-MinDate)/(MaxDate-MinDate)×100
       Rules = Rules ∪ R
     else
       break
     end if
     l = l - 1
  until l>1
end for
return Rules
```

R_1: <(c_1, t_1)> → <(c_2, t_2), …, (c_k, t_k)>

R_2: <(c_1, t_1), (c_2, t_2)> → <(c_3, t_3), …, (c_k, t_k)>

…

R_2: <(c_1, t_1), (c_2, t_2), …, (c_{k-1}, t_{k-1})> → <(c_k, t_k)>

Each rule has a confidence value which is computed by using (2). Then the rules, which have confidence values higher than a predefined confidence threshold $conf_{min}$, are selected and called frequent mobility rules.

Notice that, the aim of mining frequent mobility patterns is generating frequent mobility rules which are near to user's current movement for an accurate prediction. That means, the rules which are extracted from the recent mobility patterns should be more important than the ones extracted from the older patterns. For reaching this goal, each generated rule r_i is assigned a weighted value w_i based on temporal attribute. The weighted value of each rule are calculated as the following procedure. Let *MinDate* and *MaxDate* denote the first date and the last date in a log file of a node's mobility history, respectively. The date of the rule, which is determined through the time of the last point of the rule's tail, is called *RuleDate*. The *RuleDate* is nearer to the *MaxDate*, the weighted value is higher. Then, the weighted value is calculated by (3). According to the computation, all the rules which are extracted from a frequent mobility pattern have the same weighted value.

Algorithm 4 Mobility Prediction Algorithm

Input: Current trajectory of the user, P = <(c1, t1), (c2, t2),…, (ci-1, ti-1)>
Set of mobility rules, R
Maximum predictions made each time, *m*

Output: Set of predicted cells, *Pcells*

```
PCells = ∅ // Initially the set of predicted cells is empty
k = 1
for all rule r: <(a1,t'1), (a2,t'2) ..., (aj,t'j)>→
    <(aj+1,t'j+1), ..., (at,t't)> ∈ R do
    // check all the rules in R find the set of matching rules
    if <(a1,t'1), (a2,t'2) ..., (aj,t'j)> is contained by
    P = <(c1, t1), (c2, t2),…,(ci-1, ti-1)> and aj = ci-1
      Tdiff = 1/(|ti-1-t'j|+1)
      r.matchingscore = r.w+Tdiff
      r.score= r.confidence+r.support+r.matchingscore
      MatchingRules ←MatchingRules ∪ r
      //Add the (aj+1, r.score) tuple to the Tuples array
      TupleArray[k] = (aj+1, r.score)
      k = k+1
    end if
end for
// Now sort the Tuples array in descending order
TupleArray ←sort(TupleArray)
index = 0
// Select the first m elements of the Tuples array
repeat
  PCells←PCells ∪ TupleArray[index]
  index = index+1;
until index >= m || index >= TupleArray.length)
return Pcells
```

The improved mobility rules generation algorithm is presented in Algorithm 3.

$$weight(R) = \frac{RuleDate - MinDate}{MaxDate - MinDate} \times 100 \quad (3)$$

B. Mobility Prediction based on Temporal Constraints

In reality, the mobility history hides useful knowledge patterns that describe typical behavior of mobile nodes. These knowledge patterns, which are represented in the form of mobility rules, can be used to describe and predict the movement of mobile nodes. In order to overcome the problem of long processing time and computational expensiveness in data mining techniques, we propose that just the prediction phase is performed in an online manner whereas the frequent mobility patterns and mobility rules are discovered periodically in offline.

Moreover, this paper also reveals that if the timestamp of the last position of the rule's head (t_j) is far away from the query time (t_{i-1}), the prediction result may be inaccurate. The parameter T_{diff} is included to incorporate temporal similarity to the query time. The proposed mobility prediction algorithm is presented in Algorithm 4. The algorithm extends the one of matching the current trajectory of a node to the rule's head and finding the best matching using the time constraints. The best matched rule is temporally closest to the query time.

VI. EXPERIMENTAL EVALUATION

A. Synthetic Dataset Generation

In the scope of this paper, for experiments we built a dataset generator which describes the personal mobility behavior of a lecturer in a campus wireless network. At first, approximately 1000 frequent mobility patterns (FMPs) are automatically generated by following procedure. Each frequent mobility pattern is a movement around the bidirected graph in Figure 2. For each cell of a FMP, we assigned a predefined timestamp t_i satisfying ascending order of timestamps in the pattern. Next, we produce 5000 trajectories based on FMPs including two types. The first type consists of regular trajectories which follow a FMP and the second type consists of random trajectories which do not follow a FMP. Each regular trajectory is generated by insert some cells and corresponding timestamps between the consecutive cells of a FMP whereas each random trajectory is a random walk around the bidirected graph. Such a procedure of dataset generation is common in mobility prediction experiments [4] [7]. In this study, we generate 1000 random trajectories and 4000 regular trajectories.

From the obtained dataset, we select two subsets of about 2000 trajectories and 500 trajectories for training dataset and testing dataset respectively for each experiment. Each subset is built depending on the outlier ratio which is the ratio of the number of random trajectories to the total number of trajectories. In each experiment, the training set are utilized to discover all FMPs which are used to generate the weighted mobility rules. Meanwhile, the trajectories in the testing dataset are used in order to evaluate the prediction accuracy based on mobility rules which are extracted in previous phase.

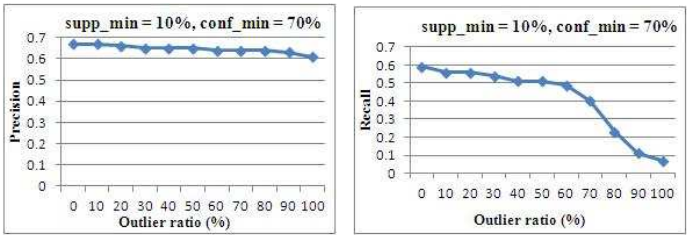
B. Experimental analysis

Traditionally, prediction accuracy is evaluated using two measures: (1) *recall measure* that is the ratio of the number of correctly predicted cells to the total number of requests. That is, the recall measure counts the "no-prediction" case (i.e., the predictor returned "no-prediction") as an incorrect prediction (i.e., the predictor incorrectly identified the next location when compared to the actual location). Hence, the larger the recall measure is, the higher prediction accuracy is. And (2) *precision measure* that is the ratio of the number of correctly predicted cells to the total number of predictions made, i.e. the "no-prediction" case is ignored. The precision measure reflects the prediction accuracy in case of incomplete dataset.

We have performed experiments on many parameters such as the outlier ratio, *o*, the minimum support threshold, $supp_{min}$ and the minimum confidence threshold, $conf_{min}$. In these experiments, we search for the best values for each parameter that make both recall and precision good.

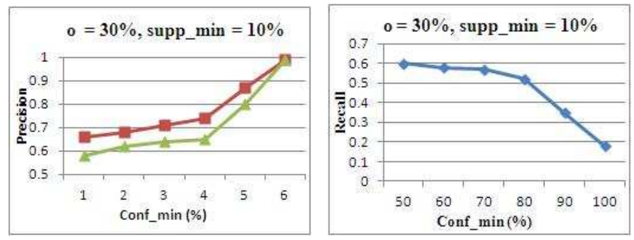
1) Effect of outlier ratio (o)

In these experiments, we vary *o* from 0% to 100% on 10% incremental steps to find how significant the regular rate of movement is to prediction accuracy; *o* = 0% means purely regular movement, whereas *o* = 100% means purely random



(a) (b)

Fig. 3. Diagrams with varied outlier ratio



(a) (b)

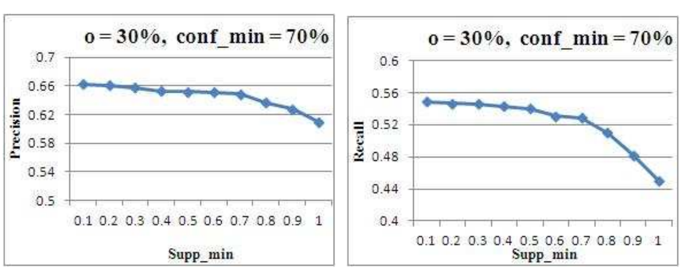
Fig. 5. Diagrams with varied minimum confidence

movement. First, we build the training dataset and the testing dataset based on the value of *o*. For example, *o* = 30%, we select 600 trajectories from 1000-trajectory random set and 1400 trajectories from 4000-trajectory regular set for training dataset. Similarly, we select 150 trajectories from 1000-trajectory random set and 350 trajectories from 4000-trajectory regular set for testing dataset. Second, we run mining algorithm on the constructed 2000-trajectory training dataset with $supp_{\min}$ = 10% and $conf_{\min}$ = 70% in order to discover all frequent mobility patterns which are used to generate weighted mobility rules. Last, the prediction algorithm is run on the constructed 500-trajectory testing dataset with respect to precision and recall measures. Figure 3 shows that the recall decreases slightly while the precision is not reduced as outlier ratio increases from 0% to 60%. However, when the outlier ratio is larger than 60%, although the precision is still not affected, the recall is strong reduced. That is due to that the probability of having some no-predictions becomes higher as *o* increases.

By considering the experimental results, the unchangeability of the precision value implies that our mining algorithm is good because most random movements are ignored. It means that all the random trajectories are eliminated in patterns discovering phase and thus they are not used in prediction phase.

2) Effect of minimum support threshold ($supp_{min}$)

We used the dataset with *o* = 30% and fixed the minimum confidence threshold at 70%. Next, the value of minimum support threshold ($supp_{\min}$) is varied to find how effective the $supp_{\min}$ value are to prediction accuracy. We vary $supp_{\min}$ from 0.1 to 1 on 0.1 incremental steps. As expected, as the $supp_{\min}$ increases, both the precision and recall values decrease (See



(a) (b)

Fig. 4. Diagrams with varied minimum support

Fig. 4). In fact, when the $supp_{\min}$ increases, the number of frequent mobility patterns decreases leading to a decreasing in number of mined mobility rules; thus, the number of correct predictions decreases.

The experiments have also showed that the increasing $supp_{\min}$ leads to a higher decrease rate in the recall values when compared to the decrease rate in the precision values. This is explained by the fact that a decrease in the number of extracted mobility rules leads to an increase in the number of no-prediction cases which are counted as incorrect predictions in the recall.

3) Effect of minimum confidence threshold ($conf_{min}$)

This experiments also used the dataset with *o* = 30% and fixed the minimum support threshold at 10%. We vary $conf_{\min}$ from 50% to 100% on 10% incremental steps. Figure 5 shows that the precision increases as $conf_{\min}$ increases. This is due to the fact that the $conf_{\min}$ value increases leading to the high quality of rules which are used for prediction. On the other hand, as the $conf_{\min}$ value increases, the number of mined rules is reduced. As a result, the number of "no prediction" cases increases. This leads to a decrease in the recall values which is illustrated by Figure 5. In this experiment, the best value of $conf_{\min}$ that make both recall and precision good is 70%.

As another results of this experiment, we have compared the precision values obtained by our approach to the values obtained by UMP-Based [4] The values of our proposed model are always better than the values of UMP-Based approach. This indicates the effectiveness of the temporal attribute in mobility prediction.

VII. CONCLUSION

This paper presents a spatio-temporal-based mobility model in wireless networks. Based on this model, we develop algorithms for discovering frequent mobility patterns and mobility rules. The mined mobility rules will be utilized in predicting the next location of a mobile node. These proposed algorithms are implemented and experimented with various parameters. Using the dataset which describes the personal mobility behavior of a lecturer in a campus wireless network, our experimental results are two-fold. First, they show that using the temporal attribute is necessary for improving the prediction accuracy. Second, they show that the prediction accuracy is not affected by regular rate of movements; it means that our mining algorithms are suitable for mobility prediction in wireless networks. Developing group mobility based on clustering mobility patterns will be our future work.

REFERENCES

[1] Shilpy Gupta, Amit Sharma and Mohit Chowdhary, "An Improved Framework for Enhancing QoS in MIPv6", International Journal on Computer Science and Engineering , ISSN : 0975-3397, Vol. 3 No. 2, 2011.

[2] Shanthy Lucy Menezes, "Optimization of handovers in present and future mobile communication networks", Doctor of philosophy in computer science, University of Texas at Dallas, 2010.

[3] Jean-Marc Francois, "Performing and Making Use of Mobility Prediction", Doctor thesis, University of Lyte, 2007.

[4] G.Yavas, D. Katsaros, O. Ulusoy and Y. Manolopoulos, "A data mining approach for location prediction in mobile environments", Journal of Data and Knowledge Engineering, Vol. 54, pp. 121-146, 2005.

[5] Andre E. Bergh and Neco Ventura, "Prediction Assisted Fast Handovers for Mobile IPv6", IEEE MILCOM 2006 Unclassified Technical Sessions, Washington D.C., 2006.

[6] Seung-Cheol Lee, Juryon Paik, Jeewoong Ok, Insang Song and Ung Mo Kim, "Efficient Mining of User Behaviors by Temporal Mobile Access Patterns", International Journal of Computer Science and Network Security, Vol. 7, No. 2, pp. 285-291, 2007.

[7] T. H. N. Vu, J. W. Lee and K. H. Ryu, "Spatiotemporal pattern mining technique for location-based service system", ETRI Journal, Vol. 30, No. 3, 2008.

[8] W. Wanalertlak, B. Lee, C. Yu, M. Kim, S.M. Park and W. T. Kim, "Behavior-based mobility prediction for seamless handoff in mobile wireless networks", Springer Science + Business Media, Wireless Netw, Vol 17, pp. 645-658, 2011.

[9] Shashi Shekhar, Ranga Raju Vatsavai, and Mete Celik "Spatial and Spatiotemporal Data Mining: Recent Advances", 2008. Available at: http://www-users.cs.umn.edu/~shekhar/research/sdm-chapter-ngdm-v2-2008.pdf

[10] T. V. Duong, D. Q. Tran and C. H. Tran, "Spatiotemporal Data Mining for Mobility Prediction in Wireless Network", National Conference of Fundamental and Applied IT Research (FAIR), Dongnai, Vietnam 2011.

[11] Cheng-Ming Huang, Tzung-Pei Hong, Shi-Jinn Horng, "Discovering mobile users' moving behaviors in wireless networks", Journal of Expert Systems with Applications, Vol. 36, pp. 10809-10814, 2009.

[12] Vincent S. T. , and Kawuu W. L., "Efficient mining and prediction of user behaviour patterns in mobile web systems", Elservier, Information and software technology, 357-369, 2006.

[13] J. W. Lee, O. H. Paek and K. H. Ryu, "Temporal moving pattern mining for location-based service", The Journal of Systems and Software 73, Elsevier, 481-490, 2004.

[14] Xiaobai Yao, "Research Issues in Spatio-temporal Data Mining", Workshop on Geospatial Visualization and Knowledge Discovery, University Consortium for Geographic Information Science, Virginia 2003.

Thuy Van T. Duong received the B.E. degree in Information Technology from Post & Telecommunication Institute of Technology, Viet Nam, in 2004 and the M.Sc. degree in Computer Science from the HOCHIMINH University of Technology, VietNam, in 2008. She is, currently, a Ph.D. candidate at Post & Telecommunication Institute of Technology, Viet Nam. Her research interests include artificial intelligence, data mining, semantic web, mobile data management, communication technology and wireless network.

She is, currently, a lecturer of Information Technology & Applied Mathematics Department, TONDUCTHANG University, HOCHIMINH city, Vietnam.

Dinh Que Tran received the M.Sc. Degree in Software Engineering from Melbourne University, Australia, in 1998 and Ph.D in Computer Science from Information Technology Institute,Vietnam, in 2000. In 2001, he works as a research fellowship at Computer and Software Engineering, Calgary University, Canada. His research interests include artificial intelligence, distributed and intelligent computing, data mining, web mining, semantic web, semantic web service, communication technology and wireless network. He is an author/co-author of many papers published in National/International Journals and Conferences.

He is, currently, Assocciate Professor of Information Technology Faculty, Post & Telecommunication Institute of Technology, Hanoi, Vietnam.