SPATIOTEMPORAL DATA MINING FOR MOBILITY PREDICTION IN WIRELESS NETWORK

Thuy Van T.Duong[1], Dinh Que Tran[2], Cong Hung Tran [3]

[1]Department of Information Technology,University of Ton Duc Thang, Ho Chi Minh City, Viet Nam

[2]Department of Information Technology, Post and Telecommunication Institute of Technology, Ha Noi, Viet Nam

[3]Department of Information Technology, Post and Telecommunication Institute of Technology, Ho Chi Minh City, Viet Nam

Abstract. Handover latency in Mobile IPv6 is the time interval during which a mobile node can not send or receive any packets. Predicting mobility of mobile node to improve the latency is one of the critical solutions that has attracted several research interests. Various approaches to mobile prediction have been proposed such as Hidden Markov models, machine learning, data mining, etc. The approach in data mining focuses on investigating the log file of node mobility history to predict the next move of mobile node. In the context of wireless network, the spatial attributes of a mobile node are changing over time, therefore time constraints between locations play an important role in studying its mobility. However, in our knowledge, most conventional studies do not consider simultaneously spatial and temporal attributes of data. In this paper, we propose a data mining based approach that utilizes spatio-temporal attributes to predict the movement of mobile node

Keywords: Data mining, prediction, handover, mobile node, wireless network

1 Introduction

In the recent years, the need of Internet connection "everywhere and at any time" has become more and more popular with IEEE 802.11 wireless local area networks (WLAN) Standards. In 802.11 WLAN components, Access Points (AP) (also known as Base Stations) are stations providing services of distributed systems and they are connected to each other via a fixed wired network. APs with parameters such as name of networks and the channel used by AP, etc., offer wireless connection with the mobile nodes such as laptops, Personal digital assistants (PDAs) or mobile phones. A handover (handoff) is the transfer of a communication from one AP to another AP as the mobile node moves [1]. During a handover, a mobile node (MN) can not send or receive any packets thus the packet loss may occur as well [2]. There has been a considerable amount of research on handover latency to reduce the delay in handover in order to achieve a seamless handover with the meaning that the user is unaware of such status [3].

In 802.11 WLAN, the mobile nodes gain access to the network through an AP by scanning all available channels with active or passive types to find an AP to associate with [3]. This task takes a lot of time and becomes the main factor contributing to the handover latency [1]. One way to solve this problem is to guess the next AP of the mobile node so as to take active scanning. Predicting the next location of the mobile node can reduce the handover delay mainly by detecting the identity of the future AP prior to the actual handover [3]. The information of the future AP can be used in handover preparation such as providing the channel information required to reduce the scanning delay. Moreover, the prior knowledge of the future network can be utilized to reduce the handover delay such as the fast mobile IPv6 protocol. Therefore, movement prediction of a MN plays a key role in optimizing handovers.

The wireless network consists of a number of APs which are called cells. Each AP transmits on one assigned channel and periodically broadcast a beacon frame on this channel. Therefore, the MN which is currently in this cell and listening to the broadcast channel will know in which cell they are now. The movement of a MN from his current cell to another cell will be recorded in a database which is called the log file of mobility history [4]. Each mobility history contains the identity (ID) of the cell which the MN is connected and the timestamp of this connection. According to [1] [3] [4] [5], data mining has been a useful approach to mobility prediction based on the log file of node mobility history. However, in our knowledge, the conventional studies do not consider spatial and temporal attributes of data simultaneously. 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 propose a data mining based approach that utilizes spatio-temporal attributes to predict the movement of mobile node. The remainder of this paper is organized as follows. Section 2 is the related works on mobility prediction. Section 3 presents the problem and its model. Section 4 introduces the mobility prediction based on spatio-temporal data mining. Section 5 is experimental results. Finally, Section 6 is conclusions and further work.

2 Related Works

Various data mining techniques have been used widely to implement the prediction algorithms. Sequential pattern mining is one of the most important techniques in discovering the trends [4] [5] and used widely to predict customer behavior in business [6] or user behavior in web systems [7], as well as to determine the location of moving objects [8] or to guess the next location of a mobile node in mobile environments [4]. It is the process of extracting certain sequential patterns whose support values exceed a predefined minimal support threshold ($supp_{min}$) [9]. The $supp_{min}$ is a pruning mechanism that is used to reduce the number of sequences to a certain level of interest and make the process more efficient [5].

Xiaobai Yao [10] indicates the need of temporal extensions of the most spatial data mining techniques. According to the work, knowledge extracted from spatio-temporal data will help us to have better prediction of spatial processes or events. The utility of spatio-temporal data mining to predict the user movement is presented in [4] [6] [8] [11] [12]. J. Kang [8] proposed a prediction model in order to improve the quality of location based services (LBS) such as traffic management or route finding system. In [12], a Moving Pattern mining algorithm for predicting individual user movement was proposed. They define the individual user as a moving object and represent the spatial location of a moving object in coordinate system with x- and y-axis. In this work, the spatial operation is used to transform moving objects' location into area for discovering significant information. And then, they apply time constraints between locations of moving objects to make valid moving sequences. Finally, the frequent moving patterns are extracted from the generated moving sequences.

Yavas et. al. [4] presented a three stage prediction algorithm based on the Apriori algorithm for predicting user movements in a PCS (Personal Communication Systems) network. They define the mobility pattern as a sequence of cells and mine frequent paths based on sequential pattern mining. According to simulation results, moderate prediction accuracy was achieved, decreasing only minimally with an increase in random mobility.

In the work [6], Taniar et. al. open up a new future for the prediction of the movement for the individual mobile user based on user movement database (UMD). Each cell in the UMD represents a coordinate which identifies the current position of a mobile user at the given time. They transform UMD into location movement database (LMD) which details the movement sequences. They uses Apriori-like algorithm and movement tree algorithm to mine movement patterns from the LMD in order to determine how mobile users travel through the location of interest from base location to another base location via multiple intermediate locations.

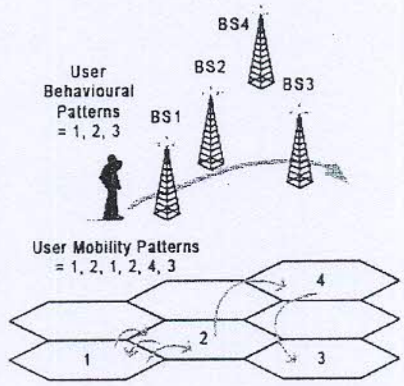

Fig. 1. Differences between user behaviour patterns and user mobility patterns [13]

However, these proposed approaches focus on the physical movements of users but do not really pay attention to user mobility from the perspective of a wireless network. What required in a wireless networks is for the mobility prediction scheme to predict the next AP through which the MN will connect to the network. In an ideal environment, the handover would be to the closest AP. However, AP overloading and anomalous propagation effects frequently result in handovers to APs other than those adjacent to the current cell. The differences between user behavior patterns and user mobility patterns are illustrated in Figure 1. The data mining based approaches [4] [6] [8] [11] [12] can be successfully utilized for location prediction of other types of moving objects, e.g., vehicles or even human but are unsuitable for location prediction for MN in wireless networks. In this paper, we determine a user's movement patterns in wireless networks through a sequence of

IDs of AP instead of a sequence of locations which are measured in 2-dimensional coordinates as in [8] [12].

3 Problem Definition

Given a directed graph G, where each vertex is a cell in the coverage region and the edges of G are formed as follows: If two cells, for example 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 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.

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 the set of all points $P = C \times T = \{(c, t) \mid c \in C \text{ and } t \in T\}$.

Definition 2: The trajectory of a 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 $1 \le j \le k$.

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, denote $B \subset A$, if and only if there exists integers $1 \le i_1 < \ldots < i_m \le n$ such that $b_k = a_{ik}$, for all k, where $1 \le k \le m$. 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: Let $D = \{S_1, S_2, \ldots, S_N\}$ be a transaction database which contains N mobility patterns. The support of pattern S is defined as

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

Definition 5: A mobility rule has the form R: $A \to B$, in which A and B are two frequent mobility patterns and $A \cap B = \varnothing$. Then, A is called the head of the rule, and B is called the tail of the rule.

Each rule R: $A \to B$ has a confidence value which is determined by the formula

$$\text{confidence}(R) = \frac{\text{support}(A \cup B)}{\text{support}(A)} \tag{2}$$

Problem definition: The problem of mobility prediction in wireless networks based on spatiotemporal data mining is defined as follows. Given a coverage region represented with directed graph G, a log file of a node's mobility history, a maximal time gap, max_{gap}, a minimum support threshold $supp_{min}$, and a minimum confidence threshold $conf_{min}$. The first phase of the problem is to generate transactional database from a log file of a node's mobility history. The second phase is to discover all frequent mobility patterns in transactional database satisfying $supp_{min}$. The third 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 a mobile node in the last part of this study.

4 Mobility Prediction Based on Spatiotemporal Data Mining

4.1 Generating Transactional Database

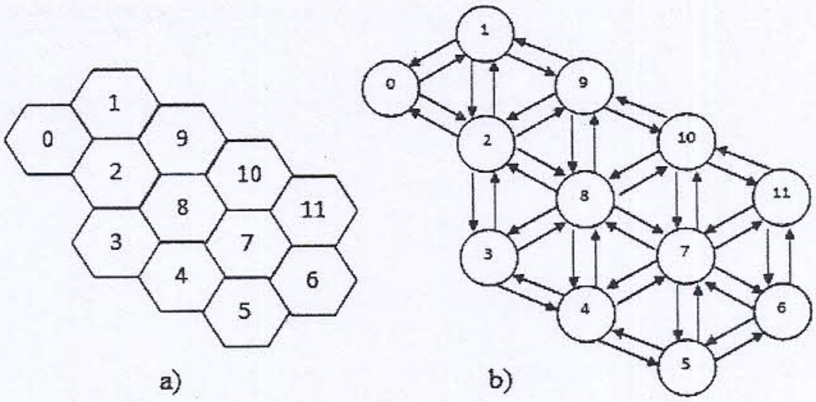

Fig. 2. The coverage region (a) and the corresponding directed graph G (b)

In our work, we assume that the mobile nodes travel around the coverage region illustrated with directed graph G (See Figure 2). The movement of a mobile node from his current cell to another cell will be recorded in a log file of mobility history. A log file of node mobility history is represented in Table 1. In this phase, we use the log file of node mobility history to generate the transactional database. In order to overcome the problem of data missing, loss of connection [5] [12], we introduce the time constraints between locations of a node to make valid transactions. Time constraints restrict the time gap between two locations in a transaction. The occurrence time point of consecutive movements are denoted by t_j, t_{j+1}. The maximal time gap, max_{gap}, is defined as follows:

$$t_{j+1} - t_j \leq max_{gap}$$

For example, if the max_{gap} is 8 hours, then the transactional database extracted from the log file of mobility history in Table 1 is presented in Table 2. Note that the time gap between the sixth and the seventh location exceeds the max_{gap} of 8 hours, thus generating a new transaction. The result shows that using temporal attribute to generate transactional database allows a more flexible handling of the transactions.

4.2 Mining Frequent Mobility Patterns

This subsection presents an algorithm, which is the modified version of the Apriori technique

Table 1. A log file of node mobility history

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

Table 2. A transactional database

ID	Transactions
1	$<(0, t_3), (2, t_4), (8, t_6), (3, t_7), (4, t_8), (7, t_9)>$
2	$<(2, t_4), (3, t_5), (8, t_6), (4, t_8), (5, t_9), (7, t_9)>$
3	$<(2, t_4), (8, t_6), (3, t_7), (4, t_8), (7, t_9), (6, t_9)>$
4	$<(9, t_3), (2, t_4), (8, t_6), (4, t_8), (5, t_9), (7, t_9)>$
5	$<(2, t_4), (8, t_6), (4, t_7), (5, t_9), (7, t_9), (6, t_9)>$

[10] [14], to discover all frequent mobility patterns (FMPs) from the transactional database D. Assume that the directed graph G has N vertices and the maximal timestamp is T. First, all frequent mobility 1-patterns are extracted from the database D according to the following procedure. To discover frequent 1-patterns, for each cell ID c_i for $1 \le i \le N$, we scan the transactional database D to find all points (c_i, t_j) for $1 \le j \le T$. Each point (c_i, t_j) is a 1-pattern. Then their support values are calculated by using Definition 4. All 1-patterns which have support values higher than a predefined minimum support threshold ($supp_{min}$) are selected and called frequent mobility 1-patterns. Let L_1 be a set of frequent mobility 1-patterns.

Second, L_1 is used to generate all candidates 2-patterns C_2. Then, all candidates 2-patterns which have support values higher than $supp_{min}$ are called frequent 2-patterns L_2. The procedure is repeated until L_k is null. The pseudo-code for discovering all the frequent mobility patterns is presented in Figure 3 and Figure 4.

Frequent Patterns Mining()

```
Input:   a transactional database, D
         minimum support threshold, supp_min
         coverage region directed graph, G
Output:  a set of frequent mobility patterns, L
// Let C_k is a set of candidates k-patterns
// Let L_k is a set of frequent mobility k-patterns
L_1 ← a set of frequent mobility 1-patterns
k = 1;
while(L_k ≠ ∅) do
{
     C_k+1 ← CandidateGeneration(L_k);
     for all mobility pattern F ∈ D
     {
        C ← {c|c ∈ C_k+1 and c ⊂ F};
        for all c ∈ C
           c.count = c.count + 1;
     }
     L_k+1 ← {c|c ∈ C_k+1 and c.count ≥ supp_min}
     L = L ∪ L_k;
     k = k+1;
}
return L;
```

Fig. 3. Frequent patterns mining algorithm

4.3 Generating Mobility Rules

This subsection presents the algorithm to generate all the mobility rules using the FMPs which are discovered in the previous phase. Let S be a FMP, all the possible mobility rules which can be generated from S are: $A \rightarrow (S - A)$ for all $A \subset S$ and $A \neq \varnothing$. Each rule has a confidence value which is computed by using Definition 5. Then, the rules, which have confidence values higher than a predefined confidence threshold $conf_{min}$, are selected and called frequent mobility rules. By using all the mined FMPs, all frequent mobility rules are generated and are used in the mobility prediction phase. Figure 5 presents the pseudo-code for generating all the frequent mobility rules.

Note that the rules, which are extracted from the recent patterns, should be more important than the rules extracted from the older movement patterns.

Definition 6: Assume that we have two rules r_1: $<(a_1, t_1), (a_2, t_2), \ldots, (a_j, t_j)> \rightarrow <(a_{j+1}, t_{j+1}), (a_{j+2}, t_{j+2}), \ldots, (a_k, t_k)>$; and r_2: $<(b_1, t'_1), (b_2, t'_2), \ldots, (b_i, t'_i)> \rightarrow <(b_{i+1}, t'_{i+1}), (b_{i+2}, t'_{i+2}), \ldots, (b_l, t'_l)>$. r_2 is a older rule iff the timestamp of the last position of it's tail is smaller than the timestamp of the last position of the tail of r_1 ($t'_l < t_k$). In this case, r_1 is known as the recent rule.

For an accurate prediction, each generated rule is assigned a weight value w_i that is determined through the timestamp of the last position of the rule's tail (t_k). The recent rules should have a greater importance than older rules. Thus, all the rules extracted from a FMP have a same weight value.

Mobility Rules Generation()

```
Input:   a set of frequent mobility patterns, L
         minimum confidence threshold, conf_min
Output:  a set of frequent mobility rules, Rules

Rules ← Ø;
for all frequent mobility pattern k-pattern
     P_k=<(c1, t1), (c2, t2), …, (ck, tk)> ∈ L_k, k≥2
{
       P_l ← P_k;
       while(l>1)
        do{
          //A ← dropping the last point of P_l i.e. A is a (l-1)-subpattern of P_l
          A ← <(c1, t1), (c2, t2), …, (c_l-1, t_l-1)>;
          conf = support(P_k)/support(A);
          if conf ≥ conf_min then
          {
              //R ← (A → P_k-A)
              R ← {<(c1, t1), (c2, t2), …, (c_l-1, t_l-1)>
                                        →<(c_l, t_l), …, (ck, tk)>}
              //all rules generated in the same year have the same weight value
              R.w = year(t_k);
              Rules = Rules ∪ R;
          }
          else
              break;
          l = l – 1;
       }
}
return Rules;
```

Fig. 4. Mobility Rules Generation Algorithm

4.4 Mobility Prediction

The frequent mobility patterns and mobility rules are discovered periodically in an offline manner while the last phase is performed online. In the last phase, the next movement of the node is predicted based on the set of the mobility rules which are generated in the third phase. This paper reveals that if the timestamp of the last position of the head of the rule (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 Figure 6. The algorithm extends the problem 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.

5 Experimental Results

To perform experiments, we assume that a mobile node moves around the coverage region directed graph G in Figure 2. The dataset is consisted of 4000 transactions with the average length of transactions is 6, outlier percentage is 30%, corruption factor is 40% and maximal timestamp is 11. From the datasets, we generate the training set of 3000 transactions and the test set of 1000

Mobility Prediction()

```
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 each rule r : <(a1,t'1), (a2,t'2) ..., (aj,t'j)> → <(aj+1,t'j+1), ..., (ai,t'i)> ∈ R
{
     // 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 for
   // Now sort the Tuples array in descending order
TupleArray ←sort(TupleArray);
index = 0;
// Select the first m elements of the Tuples array
while (index < m && index < TupleArray.length)
 do{
         PCells←PCells ∪ TupleArray[index];
         index = index+1;
}
return Pcells;
```

Fig. 5. Mobility prediction algorithm

transactions. We use the training set to discover all the FMPs and then use these FMPs to generate all mobility rules which are utilized in the prediction algorithm. The test set is used for evaluating the prediction accuracy of our approach.

These are three possible outcomes for the mobility prediction, when compared to the actual location [4]:

- The predictor correctly identified the location of the next move.

- The predictor incorrectly identified the location of the next move.
- The predictor returned "no prediction".

To evaluate performance of our proposed approach, we use two measures:

- Recall: The number of correct predictions divided by the total number of requests. The recall counts the "no prediction" case as an incorrect prediction.
- Precision: The number of correct predictions divided by the total number of made predictions. The precision skips the "no prediction" case.

Candidates Generation()

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
        P_k=<(c_1, t_1), (c_2, t_2), ..., (c_k, t_k)> ∈ L_k
{
    V(c_k) ← {v|v is a neighbor of c_k};
    for all vertex v ∈ V(c_k)
    {
  P(c_k) ← {p=(v, t_{k+1})| <(v, t_{k+1})> ∈ L_1 and t_{k+1}≥t_k};
        for all p=(v, t_{k+1}) ∈ P(c_k)
        {
            C=<(c_1, t_1), (c_2, t_2), ..., (c_k, t_k), (v, t_{k+1})>
            C_{k+1}= C_{k+1} ∪ C;
        }
    }
}
return L;
```

Fig. 6. Candidates generation algorithm

We have performed experiments on many parameters such as the maximum number of predictions made at each move of mobile node, m, the minimum support threshold, $supp_{\min}$, 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. Due to the restriction of the paper, we focus to the experiment which examines as $conf_{\min}$ increases. This is due to the fact that the $conf_{\min}$ value increases and it leads the performance impact of parameter $conf_{\min}$. Figure 7 shows that the precision increases to the high quality of rules which are used for prediction. However, as the $conf_{\min}$ value increases, the number of mined rules is reduced. The decrease in the rules leads to the

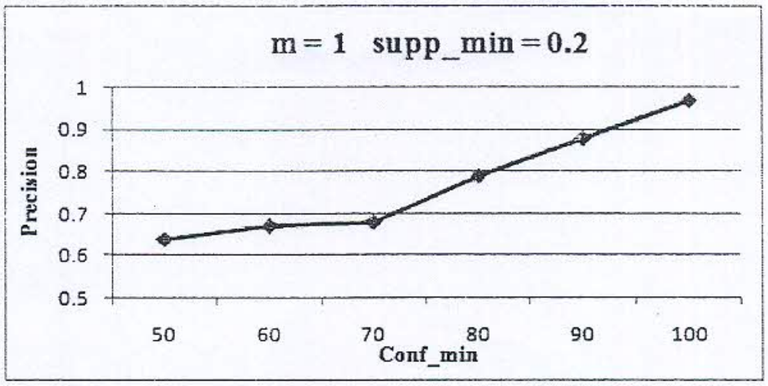

Fig. 7. Precision as a function of the $conf_{\min}$

number of "no prediction" cases increases. Therefore, the recall decreases as $conf_{\min}$ increases as shown in Figure 8. In this experiment, the best value of $conf_{\min}$ that make both recall and precision good is 70%.

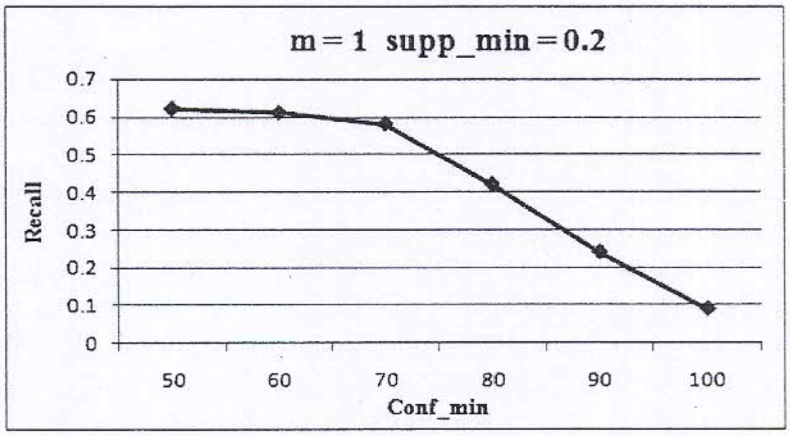

Fig. 8. Recall as a function of the $conf_{\min}$

Conclusions

The spatial attributes of a mobile node are changing over time, therefore time constraints between locations play an important role in considering its mobility. This paper introduces a mobility prediction approach based on spatiotemporal data mining. While various studies on mobility prediction focus on types of moving objects with only location, our work integrates time into rules to facilitate prediction of location of mobile nodes in wireless networks. Performing comparison evaluation with other works are currently examined and will be presented in our further work.

References

1. Jean-Marc Francois: Performing and Making Use of Mobility Prediction. Doctor thesis, Université de liège (2007)
2. Shilpy Gupta et al. : An Improved Framework for Enhancing QoS in MIPv6. International Journal on Computer Science and Engineering , ISSN : 0975-3397, Vol. 3 No. 2 (2011)
3. 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)
4. G.Yavas, D. Katsaros et al. : A data mining approach for location prediction in mobile environments. Data & Knowledge Engineering 54, page 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. D. Taniar and J. Goh: On mining movement pattern from mobile users. International journal of distributed sensor networks, ISSN: 1550-1329, 3: 69-86 (2007)
7. 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)
8. Juyoung Kang: A Frequent pattern based prediction model for moving objects. International journal of computer science and network security, vol. 10, no. 3 (2010)
9. Q. Zhao: Mining Deltas of Web Structure: Issues, challenges and solutions. PhD thesis, Nanyang Technological University, http://www.cais.ntu.edu.sg/qkzhao/pdf/FYR.pdf (2003)
10. Xiaobai Yao: Research Issues in Spatio-temporal Data Mining. Workshop on Geospatial Visualization and Knowledge Discovery, University Consortium for Geographic Information Science, Virginia (2003)
11. T. Anagnostopoulos, et al.: Path Prediction through Data Mining. IEEE International conference on Pervasive Services, Istanbul (2007)
12. 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)
13. J. Chan and A. Seneviratne: A practical user mobility prediction algorithm for supporting adaptive QoS in Wireless Networks. In Proc. IEEE Int'l Conf. Networks (ICON'99), pp.104-111 (1999)
14. Shashi Shekhar, et al.: Spatial and Spatiotemporal Data Mining: Recent Advances, http://www-users.cs.umn.edu/~shekhar/research/sdm-chapter-ngdm-v2-2008.pdf (2008)
15. Cong Hung Tran,Van T.T. Duong: Mobile IPv6 Fast Handover Techniques. In: Proc. of the 1^{3th} International Conference Advanced Communicaiton Technology, ISBN 978-89-5519-154-7, pp. 1304-1308, Korea (2011)