Comparison and Evaluation of Sequential Pattern Mining Method for Predicting Handover in Mobile IP

Tran Cong Hung, Nguyen Thi Thanh Minh, *Member, IEEE*

***Abstract*—Data mining is applied in many areas, including the telecommunication area. Telecommunication area stores a large amount of data, including the information about the user's movement. Many studies in recent years have shown that the user's movement in past hides knowledge about their movement behavior. The exploitation of spatial and temporal simultaneously will give performance better than the exploitation of one of them. This paper will analyze and evaluate the performance of sequential pattern mining in two cases: only based on spatial and based on spatial-temporal.**

***Index Terms*—data mining, mobility prediction, sequential pattern mining technique.**

I. Introduction

Mobile IP is a standard of IETF, allows users maintaining connections, and applications while moving between different networks. This is done by identifying home address of each device. When a mobile device moves outside the home network, it will send the information about the current location to home agent [1]. Home agent will receive packets is sent to the mobile device, change some of the information and forward these packets to the current location of the mobile device.

When users move from an old access point to a new access point, it will not receive any packets during this process. If delay is long; packet loss will high, and quality of service will decrease. After Mobile IP protocol had been introduced, the other protocols such as Mobile IPv6, Hierarchical Mobile IPv6, Fast Mobile IPv6 were proposed [2] [3] [4] [5]; in order to improve performance in handover process. However, these protocols such as MIPv6, HMIPv6, and FMIPv6 only improve in structure, as well as procedure and principle operation, rather than exploit the information is available in network system.

If network system has the information about the movements of user, it can register resources, reduce latency in handovers process; ensure quality of service for users. Therefore, many studies have proposed many solutions to predict the movement of users in Mobile IP network; in order to provide the information about the movement of users in future. Among these techniques are proposed, sequential pattern mining is paid attention to by many groups. However, there is no studies that compare and evaluate performance of these techniques. Therefore, the objective of the paper is analyze and evaluate performance of sequential patterns mining in two cases: only based on spatial and based on spatial-temporal.

This paper is divided into 5 sections: Section-1 introduces about Mobile IP protocol and the need for prediction in Mobile IP network, Section-2 presents the related works, Section-3 introduces sequential pattern mining in two cases: only based on spatial and based on spatial-temporal. Section-4 is the experimental results and evaluation. Section-5 is our conclusion and future works.

II. Related works

In recent years, many research groups have provide mechanisms which predict the movement of users based on their movement in past. One of them is based on neural network [6] [7] [8]. Velmurugan and Thangaraj (2013) proposed a mobility prediction model based on neural network. This study proposes a two-step method for mobility prediction. The first step suggested a mathematical model for mobility rules extraction from syslog. The second step involves clustering of extracted rules head thereby becoming the class label for a classification model.

In addition, there are many other techniques such as statistical analysis [9], using decision trees [10] [11], Markov model [12] [13]. The disadvantage of these methods is that they require a long training phase on mobile user behavior before the prediction succeeds. Moreover, the mobile user can change his behavior during the training phase or can go to a location user has never visited before, thus making the prediction ineffective.

Besides, most of the predicted mechanism is proposed based on sequential pattern mining technique [14] [15] [16] [17] [18] [19] [20] [21]. Sequential pattern mining technique proposed is based on mining the mobility patterns of users, forming mobility rules from these patterns, and finally predicting a

Manuscript received September 4th, 2013. Accepted: September 25th, 2013.

Tran Cong Hung, Ph.D. is with the Post and Telecommunications Institute of Technology, Vietnam (e-mail: conghung@ptithcm.edu.vn).

Nguyen Thi Thanh Minh, Eng. is with the Post and Telecommunications Institute of Technology, Vietnam (e-mail: thanhminhns2011@gmail.com).

mobile user's next movements by using the mobility rules. In which, there are many works only based on one of two attribute as spatial [14] or temporal [21]. However, in fact, the history of user's movement always changes over time. Therefore, the combination of two attributes will predict exactly than these techniques only based one of them [15] [16] [20].

III. SEQUENTIAL PATTERN MINING TECHNIQUE

This technique will extract a set of certain patterns from database. Then, these certain patterns are used to generate association rules. In the extraction process and generate associated rules will be identified by two parameters: the minimal confident threshold ($conf_{min}$) and the minimal support threshold ($supp_{min}$) [14] [15] [16] [19] [20].

Algorithm consists of three phases:

- User mobility pattern mining
- Generation of mobility rules
- Mobility prediction

A. Based on Spatial:

- Step 1: Mining user mobility patterns

In this method, we use a directed graph G, where the cells in the coverage region are considered to be the vertices of G. The edges of G are formed as follows: 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 B to A. We assume that the set of candidate patterns each including k cells is found in the (k-1) step run of the while loop and this set is not empty, is denoted by C_K. First all the length-k subsequences of all UAPs are generated and these subsequences are used to count the supports of the length-k candidate patterns.

Mining user mobility patterns () [14]

Input: All the UAPs in the database, D
Minimum value for support, $supp_{min}$
Coverage Region Graph, G
Output: User mobility patterns (UMPs), L

All patterns which have a length of one, C1
K=1
Initially the set of large patterns is empty, L= $\emptyset$
While ($C_k \neq \emptyset$)
{ for each UAP a $\in$ D
{ S= {s| s $\in C_k$ and s is a subsequence of a}
// S is the set of candidate length-k patterns which are also
// subsequences of UAP a
For each s $\in$ S
{ s.count = s.count + s.suppInc }
// choose the candidates which have enough support
L_k = { s | s $\in C_k$, s.count >= $supp_{min}$ }
L = L $\cup$ L_k // add these length-k large patterns to the set of all large patterns
// generate length – (k+1) candidate patterns
C_{k+1} , $\forall$ c $\in C_{k+1}$ c.count = 0
k= k+1
}
Return L.

The support is identified by using the following formula:

$$suppInc = \begin{cases} \frac{1}{1+totDist}, & \text{if pattern B is contained by UAP A} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

We can define the totDist value by means of the notion of string alignment.

After counting the supports of all the candidates, the candidates which have a support smaller than the threshold value ($supp_{min}$) are eliminated. The remaining candidates are called the length-k large patterns (L_k). Then, L_k is added to the set in which all the large patterns are maintained.

The next step in the mining algorithm is the generation of length-(k+1) candidate patterns, C_{k+1}. For this step, the CandidateGeneration() function, is used.

CandidateGeneration () [14]

Input: Length-k large patterns, L_k
Coverage Region Graph, G
Output: Length-(k+1) candidate patterns, Candidates

Candidates = $\emptyset$ // initially the candidates set is empty
For each L = $\langle l_1, l_2,\ldots,l_k\rangle$, L $\in L_k$
{// for each length-k large pattern L
// determine all the cells which are neighbor of l_k in G
N^+ = {v | there is an edge in G such as lk $\rightarrow$ v}
For each v $\in N^+$ (l_k)
{
// for each of these neighbor cells, v
// generate a candidate by attaching v to end of L
C' = $\langle l_1, l_2,\ldots,l_k, v\rangle$
// add C' to the candidates set
Candidates $\leftarrow$ Candidates U C'
}
}
Return Candidates.

- Step 2: Generation of mobility rules

Assume that we have a UMP C = $\langle c_1, c_2,\ldots,c_k\rangle$, where k>1. All the possible mobility rules which can be derived from such a pattern are:

$\langle c_1\rangle \rightarrow \langle c_2,\ldots,c_k\rangle$
$\langle c_1, c_2\rangle \rightarrow \langle c_3,\ldots,c_k\rangle$
…
$\langle c_1, c_2,\ldots,c_{k-1}\rangle \rightarrow \langle c_k\rangle$

For each mobility rule, a confidence value is calculates for each rule by using the following formula:

$$\text{conf (R)} = \frac{\langle c1,c2,\ldots,ck\rangle.count}{\langle c1,c2,\ldots,ci-1\rangle.count} \times 100 \quad (2)$$

Then, the rules which have a confidence higher than a predefined confidence threshold ($conf_{min}$) are selected. These rules are used in the next step, which is the mobility prediction.

- Step 3: Mobility prediction

Assume that a mobile user has followed a path P= $<c_1,c_2,\ldots,c_{i-1}>$ up to now. Our algorithm finds out the rules whose head parts are contained in path P, and also the last cell in their head is c_{i-1}. We call these rules the matching rules. We store the first cell of the tail of each matching rule along with a value which is calculated by summing up the confidence and the support values of the rule in an array of tuple. The support of a rule is the support of the UMP from which the current rule is generated. The tuple of this array are then sorted in descending order with respect to their support plus confidence values. While sorting the matching rules, both the support and confidence values of a rule should be taken into consideration to select the most confident and frequent rules.

```
Mobility prediction () [14]
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
      r.score= r.confidence+r.support+r.weight
      //Add the rule into the set of matching rules
      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
    While (index < m && index < TupleArray.length)
    {
        PCells ← PCells ∪ TupleArray [index]
        index = index + 1
    }
    return Pcells.
```

B. Based on Spatial-temporal:

The location factor indicates the movement of mobile users often moves sequentially every day. Besides, the time factor identifies the importance of the time when mobile user moves to the location. Therefore, in this section, two characteristics which are used to define mobility behaviors are location and time.

We set the time interval every 135 minutes. According to our work, the timestamps are illustrated in Table 1.

TABLE I
DEFINED TIMESTAMPS [15]

Timestamp	Time interval
T1	0:00 – 2:14
T2	2:15 – 4:29
T3	4:30 – 6:44
T4	6:45 – 8:59
T5	9:00 – 11:14
T6	11:15 – 13:29
T7	13:30 – 15:44
T8	15:45 – 17:59
T9	18:00 – 20:14
T10	20:15 – 22:29
T11	22:30 – 23:59

- Step 1: Mining user mobility patterns

First, all frequent mobility 1-patterns are extracted from the database D. 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 supported values are calculated. All 1-patterns which has supported values higher than a predefined minimum support threshold ($supp_{min}$) are selected and called frequent mobility 1-patterns.

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

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

For each $v \in V(c_{k-1})$, generating all points (v, t_k) satisfying $1\leq t_k \leq T$, and then $<(v, t_k)>$ is a frequent 1-pattern and $t_k \geq t_{k-1}$.Let $P(c_{k-1})=\{p=(v, t_k)|\ <(v, t_k)> \in L_1 \text{ and } t_k \geq t_{k-1}\}$ 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:

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

```
Mining user mobility patterns () [15][16]
Input:  A transactional database, D
        Minimum support threshold, suppmin
        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 ∈ Ddo
            C ←{c|c ∈ Ck+1and c ⊂ F}
```

for all c ∈ C do
c.count= c.count+ 1;
end for
end for
$L_{k+1} \leftarrow \{c|c \in C_{k+1}$ and c.count $\geq$ suppmin}
$L = L \cup L_{k+1}$
$k = k+1$
until $L_k = \emptyset$
return L.

The next step in the mining algorithm is the generation of length-(k+1) candidate patterns, C_{k+1}. For this step, the CandidateGeneration() function, is used.

CandidateGeneration () [15][16]
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 do
$V(c_k) \leftarrow \{v|v$ is a neighbor of $c_k\}$
for all vertex v ∈$V(c_k)$do
$P(c_k) \leftarrow \{p=(v, {}_{tk+1})| <(v, t_{k+1})> \in L_1$ and $t_{k+1} \geq t_k\}$
for all p=(v, t_{k+1}) ∈$P(c_k)$ do
C=<(c_1, t_1), (c_2, t_2), …, (c_k, t_k), (v, t_{k+1})>
$C_{k+1} = C_{k+1} \cup C$
end for
end for
end for
return C_{k+1}.

- Step 2: Generation of mobility rules

Assume that we have a UMP C = <c_1, c_2, …, c_k>, where k>1. All the possible mobility rules which can be derived from such a pattern are:

<c_1> → <c_2, …, c_k>
<c_1, c_2> → <c_3, …, c_k>
…
<c_1, c_2, …, c_{k-1}> → <c_k>

For each mobility rule, a confidence value is calculates for each rule by using the following formula:

$$\text{conf (R)} = \frac{<c1, c2, ..., ck>.count}{<c1, c2, ..., ci-1>.count} \times 100$$

Then, the rules which have a confidence higher than a predefined confidence threshold ($conf_{min}$) are selected. These rules are used in the next step, which is the mobility prediction.

Different from the predicted mechanism is only based on spatial, 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. 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 weighted value is calculated by following formula:

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

Mobility rules generation () [15][16]
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
P_k=<(c_1, t_1), (c_2, t_2), …, (c_k, t_k)> ∈ L_k, k≥2
do
$P_l \leftarrow P_k$
repeat
//A ← dropping the last point of P_l
A ←<(c_1, t_1), (c_2, t_2), …, (c_{l-1}, t_{l-1})>
conf = support(P_k)/support(A)
if conf ≥ $conf_{min}$ then
//R ←(A →P_k-A)
R ←{<(c_1, t_1),(c_2, t_2), …,(c_{l-1}, t_{l-1})>
→<(c_l, t_l), …, (c_k, t_k)>}
$R.w = \frac{RuleDate - MinDate}{MaxDate - MinDate}$
Rules = Rules ∪R
else
break
end if
l = l – 1
until >1
end for
return Rules.

- Step 3: Mobility prediction

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 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
r.score= r.confidence+r.support+r.weight
//Add the rule into the set of matching rules
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.
```

IV. Experimental Results and Evaluation

In this section, for experiments we build two dataset, consist:

- Training set (800 trajectories): are utilized to discover all paths which are used to generate the mobility rules.
- Testing set (100 trajectories): are used to evaluate the prediction accuracy based on mobility rules which are extracted in previous step.

There are three possible outcomes for the location prediction, when compared to the actual location:

- 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".

We use two performance measures for the evaluation of the proposed algorithm:

$$\text{Recall} = \frac{\text{The number of correctly predicted cells}}{\text{The total number of requests}} \quad (4)$$

$$\text{Precision} = \frac{\text{The number of correctly predicted cells}}{\text{The total number of predictions made}} \quad (5)$$

Besides, in this experiment, we examine the effect of $supp_{min}$ and $conf_{min}$ values on the recall and precision.

In this model, it is assumed that mobile user travels around these stations as figure-1.

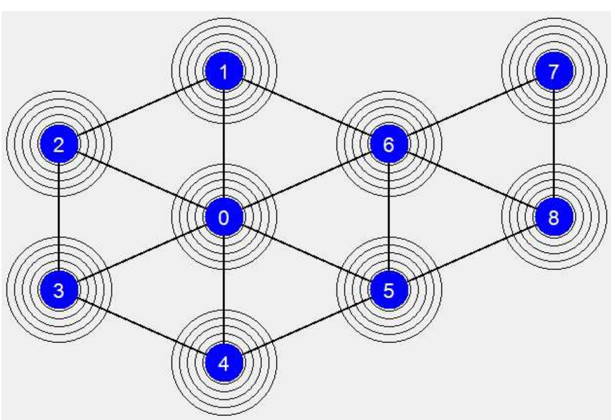


Fig. 1. A coverage region for experiments

A. Based on Spatial:

- Impact of minimum support value:

In this experiment, $conf_{min}$ is fixed at 70% and $supp_{min}$ is increased from 0.1 to 1.

TABLE II
THE PREDICTED RESULTS BASED ON THE INFLUENCE OF SPATIAL ATTRIBUTE WITH SUPPMIN

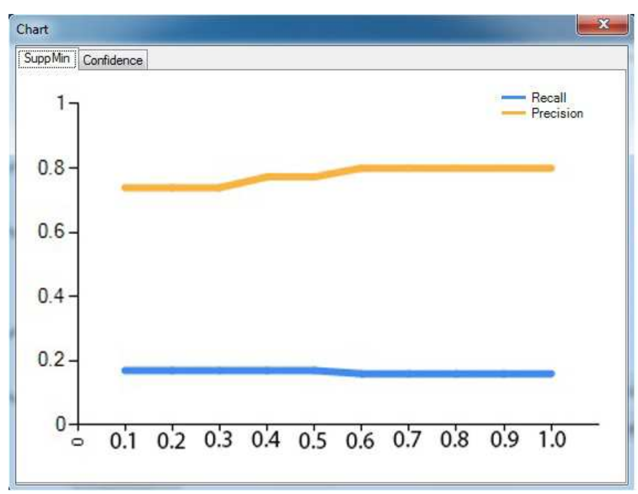
suppmin	True prediction	False prediction	No prediction	Recall	Precision
0.1	17	6	77	0.17	0.74
0.2	17	6	77	0.17	0.74
0.3	17	6	77	0.17	0.74
0.4	17	6	77	0.17	0.77
0.5	17	5	78	0.17	0.77
0.6	16	4	80	0.16	0.8
0.7	16	4	80	0.16	0.8
0.8	16	4	80	0.16	0.8
0.9	16	4	80	0.16	0.8
1	16	4	80	0.16	0.8

- Impact of minimum confident value:

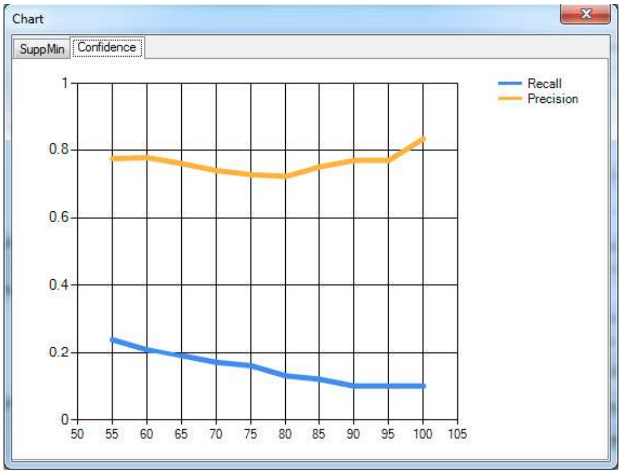
In this experiment, $supp_{min}$ is fixed at 0.1 and $conf_{min}$ is increased from 50% to 100%.

TABLE III
THE PREDICTED RESULTS BASED ON THE INFLUENCE OF SPATIAL ATTRIBUTE WITH CONFMIN

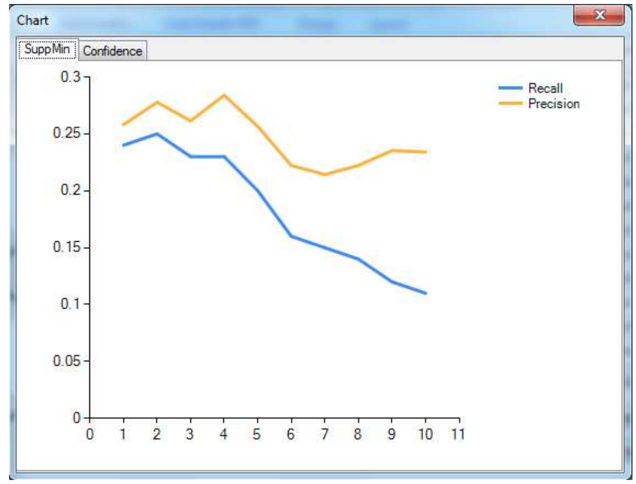
confmin	True prediction	False prediction	No prediction	Recall	Precision
55%	24	7	69	0.24	0.774
60%	21	6	73	0.21	0.777
65%	19	6	75	0.19	0.76
70%	17	6	77	0.17	0.739
75%	16	6	78	0.16	0.727
80%	13	5	82	0.13	0.722
85%	12	4	84	0.12	0.75
90%	10	3	87	0.1	0.769
95%	10	3	87	0.1	0.769
100%	10	2	88	0.1	0.833



2a. $supp_{min}$



2b. $\mathbf{conf_{min}}$



3a. $\mathbf{supp_{min}}$

Fig. 2. Recall and Precision with influence of spatial

B. Based on Spatial-temporal:

- Impact of minimum support value:

In this experiment, $conf_{min}$ is fixed at 70% and $supp_{min}$ is increased from 1 to 10.

TABLE IV
THE PREDICTED RESULTS BASED ON THE INFLUENCE OF SPATIAL-TEMPORAL ATTRIBUTE WITH SUPPMIN

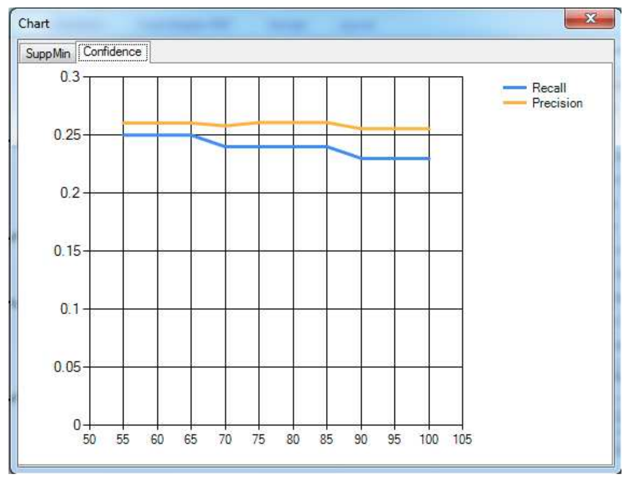
suppmin	True prediction	False prediction	No prediction	Recall	Precision
0.1	24	69	7	0.24	0.258
0.2	25	65	10	0.25	0.277
0.3	23	65	12	0.23	0.261
0.4	23	58	19	0.23	0.284
0.5	20	58	22	0.2	0.256
0.6	16	56	28	0.16	0.222
0.7	15	55	30	0.15	0.214
0.8	14	49	37	0.14	0.222
0.9	12	39	49	0.12	0.235
1	11	36	53	0.11	0.234

- Impact of minimum confident value:

In this experiment, $supp_{min}$ is fixed at 1 and $conf_{min}$ is increased from 50% to 100%.

TABLE V
THE PREDICTED RESULTS BASED ON THE INFLUENCE OF SPATIAL-TEMPORAL ATTRIBUTE WITH CONFMIN

confmin	True prediction	False prediction	No prediction	Recall	Precision
55%	25	71	4	0.25	0.26
60%	25	71	4	0.25	0.26
65%	25	71	4	0.25	0.26
70%	24	69	7	0.24	0.258
75%	24	68	8	0.24	0.26
80%	24	68	8	0.24	0.26
85%	24	68	8	0.24	0.26
90%	23	67	10	0.23	0.255
95%	23	67	10	0.23	0.255
100%	23	67	10	0.23	0.255



3b. $\mathbf{conf_{min}}$

Fig. 3. Recall and Precision with influence of spatial-temporal

V. CONCLUSION

This paper compares sequential pattern mining technique in two cases: only based on spatial and based on spatial-temporal. We build two dataset, consist: training set and testing set. After that, we make experiments and compare the results of two approaching methods on the same model. When using predicted model based on spatial-temporal, we see that the amount of right prediction is higher, and it reduces the amount of unpredicted cases. This shows that based on spatial-temporal model has accuracy higher than based on spatial model. Therefore, the time attribute of the user's movement history plays an important role in predicting the future connecting position of the mobile user.

In the future, our work is exploiting the combination of spatial and temporal in the other prediction mechanisms, to improve performance for handover process.

REFERENCES

[1] Perkins, C., (2002), *IP Mobility Support for IPv4*, RFC 3344.
[2] Perkins, C., D. Johnson, J. Arkko, (2004), *Mobility Support in IPv6*, RFC 3775.
[3] Tran Cong Hung, Nguyen Thi Thuy An, (July 2012, Canada), "Research Handover on Mobile IP", Cyber Journals: Multidisciplinary Journals in Science and Technology, ISSN 1925-2676, pp. 42-49.
[4] Soliman, H., C. Castelluccia, K. ElMalki, L. Bellier, (2008), *Hierarchical Mobile IPv6 (HMIPv6) Mobility Management*, RFC 4140.
[5] Tran Cong Hung, Duong Thi Thuy Van, (Feb 2011, Korea), "Mobile IPv6 Fast Handover Techniques", IEEE_The International Conference on Advanced Communication Technology, ICACT 2011.
[6] Velmurugan, Land and Thangaraj, P, (2013), "Mobility Prediction using Hidden Genetic Layer Based Neural Network", Life Science Journal 2013.
[7] Heni KAANICHE and Farouk KAMOUN, (April 2010), "Mobility Prediction in Wireless Ad Hoc Networks using Neural Networks", Journal of Telecomunications, Volume 2, Issue 1.
[8] Ammar Yassir, Gamal Abdel Nasir and Dr.Priyanka Roy, (2013), "Mobile Ad-hoc Networks Location Prediction by using Artificial Neural: Considerations and Future Directions", IJCTA.
[9] Marin Vukovic, Goran Vujnovic, Darko Grubisic, "Adaptive User Movement Prediction for Advanced Location-aware Services".
[10] Vincent S. Tseng, Kawuu W. Lin, (2006), "Efficient mining and prediction of user behavior patterns in mobile web systems" Information and Software Technology 48, pp. 357–369.
[11] Vincent S. Tseng, Eric Hsueh-Chan Lu, Cheng-Hsien Huang, "Mining Temporal Mobile Sequential Patterns in Location-Based Service Environments".
[12] Christine Cheng, Ravi Jain, Eric van den Berg, (2002), "Location Prediction Algorithms for Mobile Wireless Systems".
[13] R.V. Mathivaruni and V.Vaidehi, (2008), "An Activity Based Mobility Prediction Strategy Using Markov Modeling for Wireless Networks", World Congress on Engineering and Computer Science 2008.
[14] Yavas, Gokhan., Dimitrios Katsaros ,Ozgur Ulusoy, Yannis Manolopoulos, (2005), *A data mining approach for location prediction in mobile environments*, Data & Knowledge Engineering , pp. 121–146.
[15] Thuy Van T.Duong and Dinh Que Tran, (2012), "An Effective Approach for Mobility Prediction in Wireless Network based on Temporal Weighted Mobility Rule", International Journal of Computer Science and Telecommunication.
[16] Thuy Van T.Duong, Dinh Que Tran, Cong Hung Tran, (August 2011), "Spatiotamporal Data mining for Mobility prediction in Wireless Network", Proceeding of the Fifth National Conference-Fundamental and Applied Information Technology Research (FAIR)-NAFOSTED, pp. 227-238.
[17] Rachida Aoudjit, Malika Belkadi, Mehammed Daoui, Lynda Chamek, Sofiane Hemrioui and Mustapha Lalam, (2013), "Mobility Prediction Based on Data mining", International Journal of Database Theory and Application.
[18] Walisa Romsaiyud, Wichian Pre mchaiswadi, Nucharee Pre mchaiswadi, (2012), "An Autonomous Group Mobility Prediction Model for Simulation of Mobile Ad-hoc through Wireless Network", Journal of Wireless Networking and Communications 2012, pp. 126-135.
[19] Thi Hong Nhan Vu, Jun Wook Lee, and Keun Ho Ryu, (2008), "Spatiotemporal Pattern Mining Technique for Location-Based Service System", ETRI Journal, Volume 30, Number 3.
[20] Cong Hung Tran, Thuy Van T. Duong and Dinh Que Tran, (April 2012), "Future Location Prediction in Wireless Network Based on Spatiotemporal Data Mining", Journal of Communication and Computer 9, ISSN 1548-7709 (Print), ISSN 1930-1553 (Online), Volume 9, Number 4, pp. 437-480.
[21] Yong-Hyuk Kim and Yourim Yoon, (2013), "Context Prediction of Mobile Users Based on Time-Inferred Pattern Networks: A Probabilistic Approach".

TRAN CONG HUNG was born in Vietnam in 1961

He received the B.E in electronic and Telecommunication engineering with first class honors from HOCHIMINH University of technology in Vietnam, 1987.
He received the B.E in informatics and computer engineering from HOCHIMINH University of technology in Vietnam, 1995.
He received the master of engineering degree in telecommunications engineering course from postgraduate department Hanoi University of technology in Vietnam, 1998.
He received Ph.D at Hanoi University of technology in Vietnam, 2004.
His main research areas are B – ISDN performance parameters and measuring methods, QoS in high speed networks, MPLS.
He is, currently, Associate Professor PhD. of Faculty of Information Technology II, Posts and Telecoms Institute of Technology in HOCHIMINH, Vietnam.

NGUYEN THI THANH MINH was born in Vietnam in 1987

Obtained BE in Information Technology from Post and Telecommunication Institute of Technology (PTIT), Vietnam, 2010.
Will receive Master degree in PTIT, 2013, major in Networking and Data Transmission.
She is, currently, IT Engineer at Vietnam Posts and Telecommunications Group (VNPT), 2011.