IJCTA, 9(36), 2016, pp. 481-489

An Efficient User Behavior Pattern Mining and Prediction in Mobile Web Systems

R. Karpagam* **and S. Suganya****

Abstract : The development of web and wireless technologies allowed the mobile users to access the various kinds of services by mobile devices at anytime and anywhere. Mobile User behavior discovery can highly benefit the changes on system performance and Quality of services (QoS).In this paper an efficient user behavior pattern mining and prediction framework is proposed with three phases. Similarity interference model is adopted for measuring the similarity among store, item and page. Personal Mobile Commerce Pattern Mine (PMCP-Mine) algorithm and GIT- Tree is applied to discover the mobile user behavior pattern. Finally prediction of possible mobile user behaviors is obtained by using the longest chain subsequence algorithm. The performance evaluation produces the efficient result.

Keywords : Pattern Mining, prediction, Mobile Commerce, Longest chain subsequence.

1. INTRODUCTION

M-commerce industry and *e*-commerce are different areas growing rapidly in recent years. Each and every day millions of transactions take place in M-commerce. The benefits of modeling the behavior patterns of users in the mobile systems provide the smart access for users and financial profit for mobile service providers such as advertising. In the mobile web environments, the mobile users may request various kinds of services and applications by cellular phone, PDA, or notebook from arbitrary locations at any time via GSM, GPRS or wireless networks. Obviously, the behavior pattern, in which the location and the service are inherently coexistent, of mobile users becomes more complex than that of the traditional web systems.

User getting desired information in a short time is one of the promising applications, especially in the mobile environments, where the users do not have much time to surf the web pages. Figure 1 shows the taxonomy for mobile commerce. The first category is about similarity measure *i.e*. measuring the relationship of stores, items and page rank. Second one is about Prediction. The two types of prediction are vector based and pattern based prediction. Based on moving direction and velocity, the vector based prediction, predicts the next location of an object whereas pattern based prediction, predicts user movement and purchase transactions.

Similarity computation method named SimRank is proposed [1]to solve the problem of measuring "similarity" of objects arises. Similarity inference model for measuring the similarities among stores and items, the idea of vector based prediction is to predict the next location of an object according to its moving direction and velocity. Vector based predictions assume that the predictive mobile behaviors of a user can be represented by mathematical models based on his recent movement in form of geographic information. Jignesh M. Patel et al. [2] has proposed an indexing method, called STRIPES. Moving object databases are required to support queries on a large number of continuously moving objects.

* Associate Professor, School of Computer Studies –PG, Rathnavel Subramaniam College of Arts and Science, Coimbatore
** Associate Professor, Department of Research in Computer Science, Rathnavel Subramaniam College of Arts and Science, Coimbatore

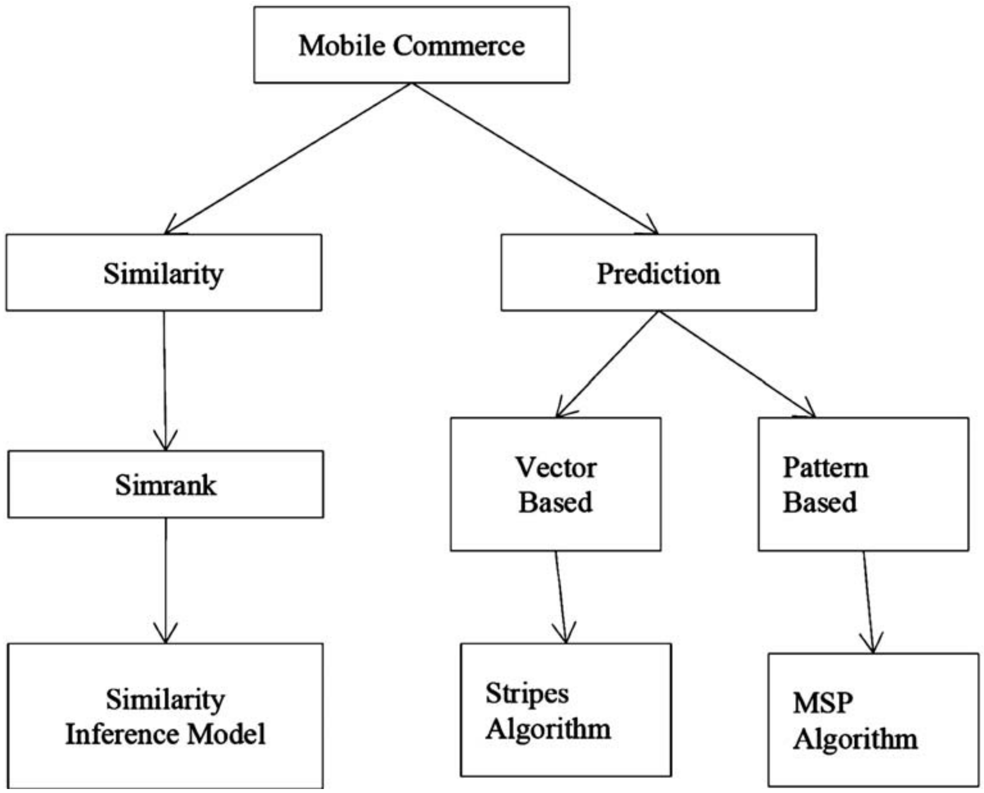

Figure 1: Mobile commerce categories

The idea of pattern based prediction is to predict the user's movements and also purchase transactions. It captures semantic patterns that match the user's recent mobile behaviors well. Pattern based predictions are more precise than vector based prediction. Yun et al. [3] has proposed a method for mining mobile sequential pattern (MSP) by taking moving patterns and purchase transactions of customers into consideration. Here they devise three algorithms (algorithm TJLS, algorithm TJPT, and algorithm TJPF) for determining the frequent sequential patterns, which are termed large sequential patterns in this work, from the mobile transaction sequences.

In this approach an efficient mining and prediction technique is proposed in mobile commerce. Similarity mechanism is utilized to obtain the similarity among the store, item and page rank. In prediction technique the pattern based approach is applied for mining the sequential pattern. Transaction set Join with Pattern Family Algorithm (TJPF) is applied for obtaining the frequent transaction and GIT tree is used for mining the frequent item set. Longest chain subsequence algorithm is used to predict the user behaviors.

2. RELATED WORKS

This section describes the related studies on mobile behavior prediction which is a major classification of mobile commerce. Previous research work on prediction and some similarity measure techniques are discussed here. Sequential pattern mining was first proposed in [4] for obtaining the inter transaction patterns in the traditional retailing. After that, many methodologies on sequential pattern mining have been developed

Yufei Tao et al [5] proposed a prediction framework based on vector based prediction methodology. This framework is designed to monitor and index the moving objects. A filter-refinement mechanism is adopted to process the query after indexing the object location. A novel recursive motion function is adopted to supports a wide class of non-linear motion patterns. It postulates the specified motion of each object by verifying its locations at recent timestamps. An efficient indexing scheme is also presented to ease the process of predictive queries without false misses. However, all this approach does not predict the next location accurately.

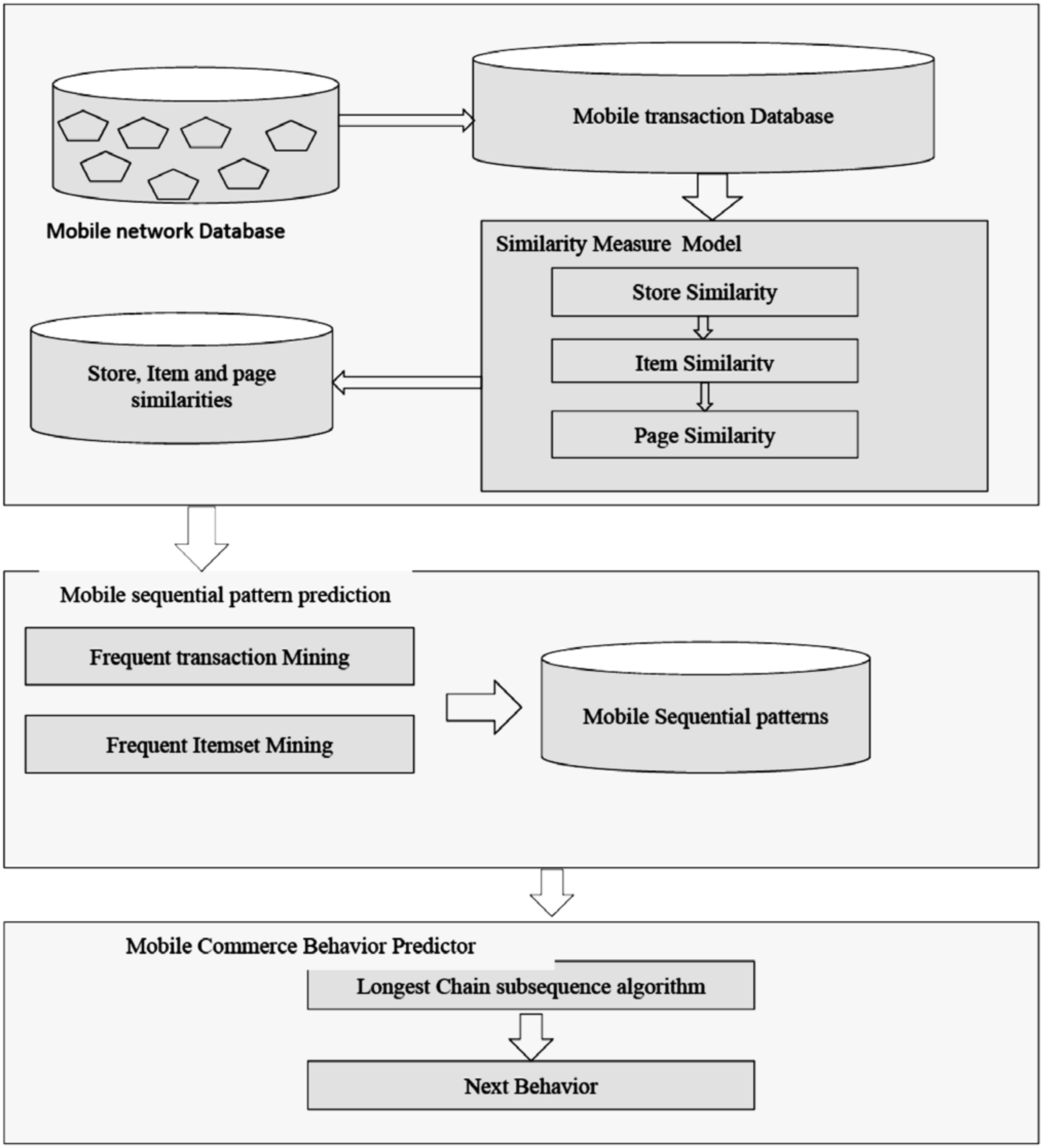

Figure 2: Proposed Framework

Bai-En Shie et al [6] proposed the two tree-based methods, namely mining high Utility Mobile Sequential Patterns with a tree-based Depth First Generation strategy (UMSPDFG) and mining high Utility Mobile Sequential Patterns with a tree-based Breadth First Generation strategy (UMSPBFG). During the mining process, the pattern length will vary. This is the difference between two methods. Mobile Transaction Sequence Tree (MTS-Tree)algorithm is utilized by both method to summarize the data about locations, items, paths and utilities in mobile transaction databases.

Ming-Syan Chen et al [7] explored a novel data mining capability for mining access patterns in a distributed information-providing environment. In this approach the documents or objects are linked together for interactive access which provides an efficient access between highly correlated objects and leads to put advertisements in proper places, better user classification and behavior analysis, etc. Maximal forward algorithm is applied to convert the original sequence of log data into a set of traversal subsequences. Then a large reference sequences algorithm is derived to determine the frequent traversal patterns from maximal forward references. However this system only filtered the effect of some backward references.

Dong Xinet al [8] addressed the compressing frequent-pattern sets. A distance measure is introduced between two frequent patterns. Second, we define a clustering criterion, with which, the distance between the representative pattern and every other pattern in the cluster is bounded by a threshold δ. The objective of the clustering is to minimize the number of clusters (hence the number of representative patterns). Finally, we show the problem is equivalent to set-covering problem, and it is NP-hard *w.r.t.* the number of the frequent patterns to be compressed. We propose two greedy algorithms: the first one, RPglobal, has bounded compression quality but higher computational complexity; whereas the second one, RPlocal, sacrifices the theoretical bound but is far more efficient.

3. PROPOSED METHODOLOGY

This section describes the design and prediction method framework for mobile commerce mining. The main objective of this approach is to predict the mobile users' movements and purchase transactions in mobile commerce. The proposed framework contains three modules. (*i*) Similarity Inference Model- The similarities among store, item and page are measured using the similarity inference model. (*ii*) Mobile sequential pattern (MSP) mining- The sequential patterns are mined from the mobile transaction sequences. (*iii*) User Behavior Prediction model- for predicting the mobile user behavior efficiently.

Similarity Inference Model (SIM)

The mobile information such as user identification, stores, item purchased etc. is stored in mobile transactional database, when mobile users move between the stores. The mobile web system has the distributed property, so the database contains different logs recording with different parts of the user's activities. Generally the mobile web logs can be classified into two types:logs that contain users' movement, and users' service requests. These logs must be Collected and integrated into one dataset before conducting the data mining methods, for efficient access.

In this approach, the similarities among the store, item and page is measured. The similarity score is obtained by deriving SIM which assigns a similarity score for every pair of stores or items or page.SIM assigns them a similarity score. In order to obtain the store similarity, two stores(ρ_x, ρ_y)are considered in this approach. These two stores are similar if they provide similar items. The store similarity is achieved by computing the average similarity of item set provided by $store_x$, $store_y$. The SIM (ρ_x, ρ_y)

$$\text{SIM}(\rho_x, \rho_y) = \frac{\sum_{\varphi \in \tau_{\rho_x}} \text{Max SIM}(\varphi, \tau_{\rho_x}) + \sum_{\sigma \in \tau_{\rho_y}} \text{Max SIM}(\sigma, \tau_{\rho_y})}{|\tau_{\rho_x}| + |\tau_{\rho_y}|}$$

The above equation describes that the store similarity can be obtained by averaging all similar item pair where Max SIM(r, R) = $\text{Max}_{r'\,R}$ SIM(r, r') which denotes the maximal similarity between r and the element in R. similarly two items (i_a, i_b) are consider to identify the item similarity. Similarity SIM(i_a, i_b) is computed by calculating the average dissimilarity of store sets that provide i_a and i_b. Therefore, SIM (i_a, i_b)) is defined as

$$\text{SIM}(i_a, i_b) = 1 - \frac{\sum_{\varphi \in \tau_{\rho_x}} \text{Max SIM}(\varphi, \tau_{\rho_x}) + \sum_{\sigma \in \tau_{\rho_y}} \text{Max SIM}(\sigma, \tau_{\rho_y})}{|\tau_{\rho_x}| + |\tau_{\rho_y}|}$$

Table 1
Similarities of store, item and page

Store	*Item*	*Page*
ρ_1, ρ_2, ρ_6	i_1, i_2, i_3	$A \rightarrow B \rightarrow C$
ρ_2, ρ_4	i_3	$B \rightarrow E \rightarrow H$
$\rho_3,$	i_5	$A \rightarrow C \rightarrow D \rightarrow E$
ρ_4, ρ_1, ρ_5	i_6, i_2	$D \rightarrow E \rightarrow F$

Table 1 describes the similarities among the store, item and pages. The store and item similarities are computed by applying the above formula and the pages are extracted from the log files directly. In this first phase the similarities are computed and the following section gives the detailed description about the mining of mobile commerce patterns efficiently.

Mobile Commerce Pattern (MCP) mining

The mining procedure of MCP is based on PMCP-Mine algorithm which has three major steps such as (*i*) Frequent Transaction mining (*ii*) Frequent Item set mining and (*iii*) MCP mining. The Max-miner algorithm and the GIT tree is adopted in this phase which is described below.

Frequent Transaction mining: Max- miner

Algorithm for generating frequent transaction in mobile transaction database

```
MAX-MINER(Data-set D)
FT: Frequent Transaction
C_g: Set of candidate group
T: Set of transaction
begin
set C_g ← {}
whileC_g is non-empty do
Scan D
Count the support of all candidate group in C_g
for each g ∈ C_g
sethd(g) ∪ tl(g) is frequent
FT ← FT ∪ {hd(g) ∪ tl(g)}
Set of Candidate Groups C_gnew ← {}
for each g ∈ C_g
sethd(g) ∪ tl(g) as infrequent
FT ← FT ∪ {gen_subnodes(g, C_gnew)}
C_g ← C_g new
if (any transaction with a proper superset in FT)
Then
Remove transaction from F
if (any group G such that hd(g) ∪ tl(g)has a superset in FT)
Then
Remove the group from C_g
Return F
```

The Max-Miner approach is utilized to obtain the frequent transaction from the mobile transaction database which can be described using Rymon's generic set enumeration tree search framework [9].The idea used in max-miner is to expand sets over an ordered and finite item domain. The specific ordering set on the item domain affects the parent/child relationships in the set-enumeration tree but not its completeness. Max-Miner uses pruning based on subset infrequency, as does Apriori, but it also uses pruning based on superset frequency. in order to support pruning efforts, each node in the set enumeration tree is called as candidate group. A candidate groupG consists of two item sets such as head and tail denotes as *hd*(G) and *tl*(G)respectively where head represents the item set enumerated by the node and is an ordered set and tail contains all items not in *hd*(G).While counting the support of a candidate group G, the support of item sets *hd*(G), *hd*(G) $\cup$ *tl*(G) and *hd*(G) $\cup \{i\} \; \forall \; i \in$ *tl*(G) has to be computed

Generating Initial Group :

```
Gen_initialgroup(Database D, set of candidate groups C_g)
F1 : set of frequent 1 transaction
FT 1: frequent 1- Transaction
begin
scan D to obtain F1
set an ordering on the transaction in F1
for each transaction I in F1
setG as new candidate with hd(G) ={i}
C_g ← C_g ∪{G}
return the transaction in F1
```

Generating Sub Nodes:

```
m: greatest item in tl(G)
begin
if(hd(G) ∪ {i} is infrequent)
remove i from from tl(G)
for each I ∈ tl(G)
setG′ as new candidate with hd (G′) = hd(G) ∪{i}
C_g ← C_g ∪{ G′}
return hd ∪ {m}
```

The max-Miner approach provides the frequent mobile transaction from the mobile transaction database. The following section describe the frequent item *i.e*. frequent pattern from this transaction.

Frequent Item set mining: GIT-tree

The frequent Item set or pattern from the frequent mobile Transaction is extracted by applying the GIT-tree. It is derived from IT tree [10].

Let D be the frequent mobile transaction and I be a set of items. Each transaction contains a unique identifier *t_id* with a set of items. The term T used in this approach denotes the set of *t_ids*. When a set P $\subseteq$ I then it is known as item set similarly if Q $\subseteq$ I then it is called as *t_id* set. Suppose that hierarchical database is a graph in I $\cup$ J, where J denotes the general set of items that derive from I. An arc in graph G represents the is-a relation, means that if an arc starts from j to i, then j is parent of i. The general association rule for a set of transaction D and a hierarchy tree G is in the form of P $\rightarrow$ Q – P, where P,Q $\subset$ I $\cup$ J, $\varnothing \neq$ P $\subset$ Q denote there is no item in Q and it is the root node of any item. The support of rule is sup(Q). The confidence of rule, conf(P $\rightarrow$ Q – P) = sup(Q)/sup(P). Let ms(P) be the minimum support of an item $x \in I \cup J$. An item set P = $\{P_1, P_2, \ldots, P_n\}$, where $P_i \in I \cup J$ and $1 \leq i \leq k$, is frequent if the support of P is greater or equal the smallest value of items in P.

$$\text{support(P)} \geq \min_{p_i \in P} \text{min\_sup}(p_i)$$

Where $\min_{p_i \in P} \text{min\_sup}(p_i)$ denotes the minimum support of an item. GIT tree contains three fields such as item set; set of transaction contains item set and minimum support which are denoted as P, *t*_idset and min_sup respectively.

MST: Minimum support table
IP: Item and its parent node in G
S: Sort(MST)
R = R—gen (S, D, IP)
$\mathbf{F_g}$ **:** First level of GIT tree
begin :
Enumerate_gen (F_g)
For each node P do
(P*T_idset (p) $\in F_g$
min_sup(P)
Set $F_g = \varnothing$
For each node Q do
(Q*T_idset (Q) $\in F_g$
(min_sup(Q)
if $\forall\ p \in P', \neg \ni q \in P'$: parent(p) = q then
T = *t_id* set(P) $\cap$ *t_id* set(Q)
min_sup(P′) = min(min(P), min(Q))
if |T| $\geq$ min(P′) then
$F_g = F_g \cup \begin{Bmatrix} P' * T \\ \text{min\_sup}(P') \end{Bmatrix}$
Enumerate_gen (F_g)

The above GIT tree procedure mine the frequent mobile E-commerce pattern effectively.

Longest chain subsequence algorithm

The final phase of the proposed work is to discover PMCPs and predict the users' future mobile commerce behaviors. It is most termed problems to determine the similarities of two sequences a and bin pattern matching. The subsequence of maximal length common to both sequences can be measured by LCS string comparison.

In general there two sequencesa = $a_1, a_2, \ldots, a_n$ and $b = b_1, b_2, \ldots, b_n$ is a subsequence of a only if increasing sequence $<h_1, h_2, \ldots h_n>$ of a such that, for all i = 1, 2, .., lit has $a_{hi} = b_i$. In a and b sequences if b is a subsequence of both a and m then b is common subsequence of a and m. Then calculate longest common subsequence (LCS) for given two sequences $a = a_1, a_2, \ldots, a_n$ and $m = m_1, m_2, \ldots, m_n$ of page-visits.

Longest Common Sequence classification procedure

1. Let us consider two sequences $P = P_1, P_2, \ldots, P_n$ and $Q = Q_1, Q_2, \ldots, Q_n$ from frequent set.

2. From the prediction list, the system must find the cluster that match with user request using following steps.
3. LCS (P_i, Q_i) denotes longest common subsequence set.
4. Find longest subsequences common to P_i and Q_i the elements p_i and q_i are compared.
5. If equal, then the sequence LCS (P_{i-1}, Q_{j-1}) is extended by that element, *xi*.
6. Else two longest common sequences, LCS(P_i, Q_{j-1}) and LCS (P_{i-1}, Q_j) are retained.

If two sequences do not match then, it is not used for predicting the user's next behaviors. If the two sequences are of the same length but not identical, then they are retained. They are the two candidates for the behavior prediction. If first sequence has a larger support then it is used to predict the user's next behaviors.

4. EXPERIMENTAL RESULT

This section describes the performance evaluation of this approach. The performance metrics such as execution time, precision, recall and f-measure are evaluated and their result is given in below figures. The experiment of this approach is implemented in java using NetBeans IDE.

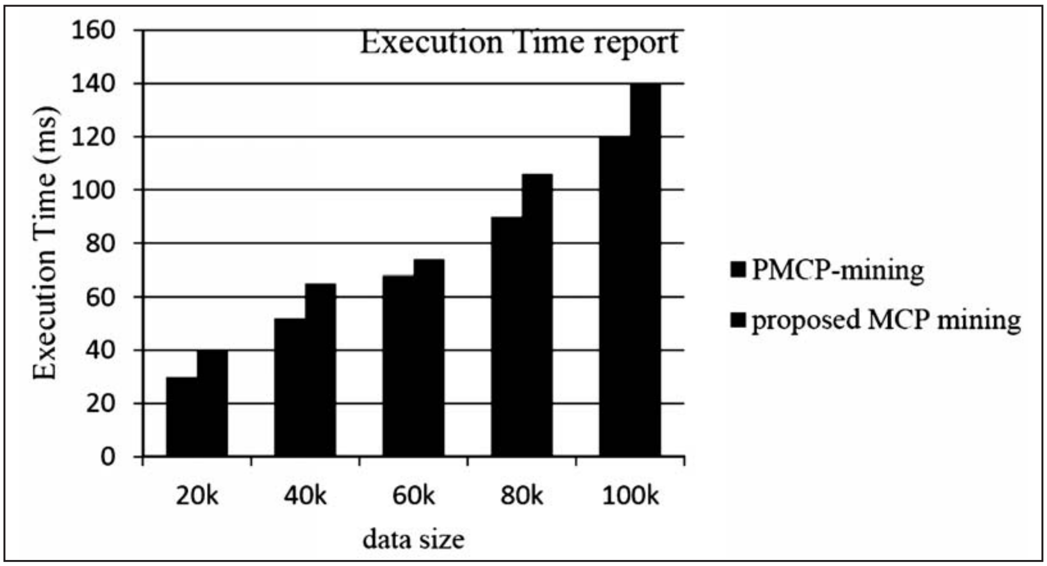

Figure 3: Comparative result of execution time

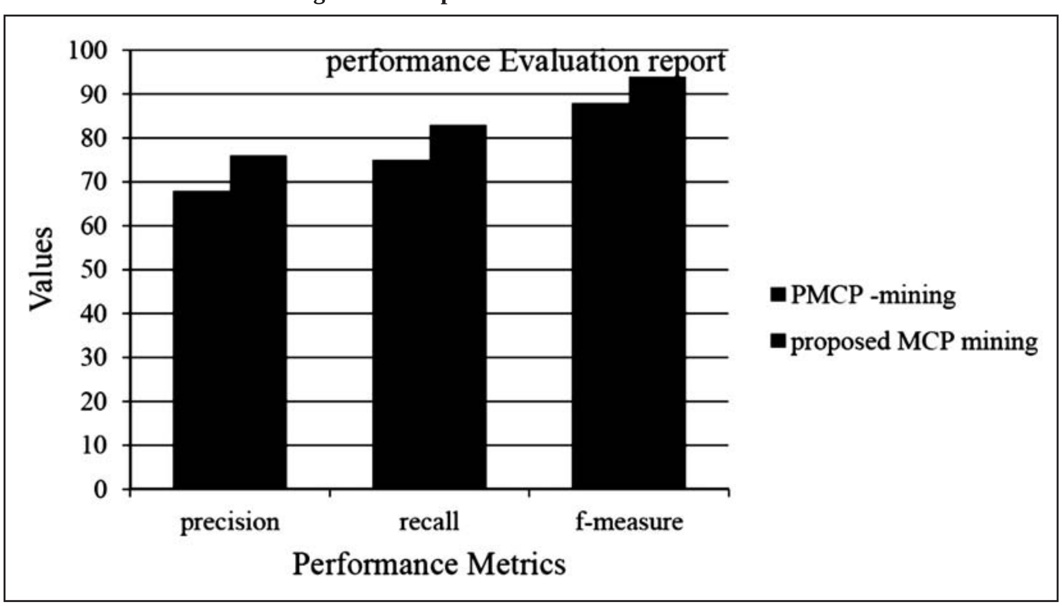

Figure 4: Comparative result of precision, recall and *f*-measure

Figure 4 and gives the comparative result of execution time and precision, recall, *f*-measure result. compared to the existing PMCP approach the proposed work obtain the efficient result.

5. CONCLUSION

This paper proposed a novel approach for mining and predicting the mobile user behavior. Similarity interference model is adopted for measuring the similarity among store, item and page. Personal Mobile Commerce Pattern Mine (PMCP-Mine) algorithm and GIT- Tree is applied to discover the mobile user behavior pattern. Finally prediction of possible mobile user behaviors is obtained by using the longest chain subsequence algorithm. The experimental results show that the framework MCE achieves a very high precision in mobile commerce behavior predictions. Besides, the prediction technique MCBP in our MCE framework integrates the mined PMCPs and the similarity information achieves superior performance in terms of precision, recall, and F-measure.

6. REFERENCE

1. G. Jeh and J. Widom, “SimRank: A Measure of Structural- Context Similarity,” Proceeding International Conference on Knowledge Discovery and Data Mining, pp. 538-543, July 2002.
2. J. M. Patel, Y. Chen, and V. P. Chakka, “Stripes: An Efficient Index for Predicted Trajectories,” Proceeding ACM SIGMOD Conference on Management of Data, pp. 635-646, June 2004.
3. C. H. Yun and M. S. Chen, “Mining Mobile Sequential Patterns in a Mobile Commerce Environment,” IEEE Transaction on Systems, Man, and Cybernetics, Part C, vol. 37, no. 2, pp. 278-295, March 2007.
4. Zaki, Mohammed J., and Ching-Jui Hsiao. “Efficient algorithms for mining closed item sets and their lattice structure.” Knowledge and Data Engineering, IEEE Transactions on 17, no. 4 (2005): 462-478.
5. Tao, Yufei, Christos Faloutsos, DimitrisPapadias, and Bin Liu, “Prediction and indexing of moving objects with unknown motion patterns”, In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 611-622. ACM, 2004.
6. Shie, Bai-En, Hui-Fang Hsiao, Vincent S. Tseng, and S. Yu Philip. “Mining high utility mobile sequential patterns in mobile commerce environments.” In Database Systems for Advanced Applications, pp. 224-238. Springer Berlin Heidelberg, 2011.
7. Chen, Ming-Syan, Jong Soo Park, and Philip S. Yu. “Efficient data mining for path traversal patterns.” Knowledge and Data Engineering, IEEE Transactions on 10, no. 2 (1998): 209-221.
8. Xin, Dong, Jiawei Han, Xifeng Yan, and Hong Cheng. “Mining compressed frequent-pattern sets.” In Proceedings of the 31st international conference on Very large data bases, pp. 709-720. VLDB Endowment, 2005.
9. “Energy Efficient Two-Phase Sensing for Cooperative Spectrum Sensing in Cognitive Radio Ad hoc Networks”in Central government NISCAIR, Journal of Scientific & Industrial Research (JSIR), New Delhi, india in September 2016 issue.