**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88


This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

# Frequent Pattern Mining Approach for a Mobile Web Service Environment Using Service Utility

**Krishna Kumar Mohbey**

Central University of Rajasthan, Department of Computer Science, Ajmer, India, kmohbey@curaj.ac.in

Received 23 October 2017; received in revised form 12 August 2018; accepted 26 September 2018

**Abstract**

Mobile web services pattern mining is an emerging field today, in which utility is playing an important role. Utility may be the profit, cost or price of an item. In the case of mobile web services, accessed preference is considered as a utility. With the help of utility mining, one can extract highly interesting frequent patterns of mobile web services. In previous related studies, most of the approaches use utility as an essential parameter to discover interesting patterns, but they also generate a large number of uninteresting patterns too. Another problem is related to computational time; because no filtration is applied and computational time is too much. In this paper, an efficient approach, Utility Based Frequent Pattern Mining, is proposed. It extracts utility based frequent patterns with high filtration in less computing time. The experimental results show that the proposed approach has good performance in terms of execution efficiency and memory utilization.

**Keywords:** Frequent service patterns, Knowledge discovery, Mobile services, Sequence utility, Sequential pattern mining

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

# 1 Introduction

Knowledge discovery is used to extract useful rules or patterns from data sets. To discovered knowledge from massive data, various data mining techniques are used. Frequent pattern mining is one of the more important approaches for generating hidden knowledge from massive data. For frequent pattern mining, Apriori was proposed [1], and it has been revealed that the algorithm has limitations, multiple scans and generating a large number of candidate itemsets. To solve this problem, Han et al. proposed the FP-Growth algorithm [10]. This can discover all frequent patterns with only two database scans. Frequent pattern mining can be applied to transactional data as well as sequential data [2]. There are different application areas of frequent pattern mining. Frequent pattern mining can be applied to mine knowledge from the mobile service accessed sequences. As we know, smartphones are widely used by people. Smartphones are used to perform online transactions, retrieving information, social messaging, video calling, chatting, etc. Services which are accessed by smartphones or laptop devices are called mobile web services. These web services are lightweight applications used to perform a specific task such as booking a ticket through a mobile app or sending a message through WhatsApp. A particular user may access series of services at different times at different locations or a single location. To extract the interesting pattern of services, data mining techniques are used. By sequential pattern mining [2], [17] web services sequence can be extracted. These sequences are helpful to find the behavior of a specific user. The generated web service patterns are used in different fields like behavior analysis of users, finding most accessed services occurrences, planning a new service, promoting business, etc. Figure 1 shows the simple scenario for mobile web service sequence generation.

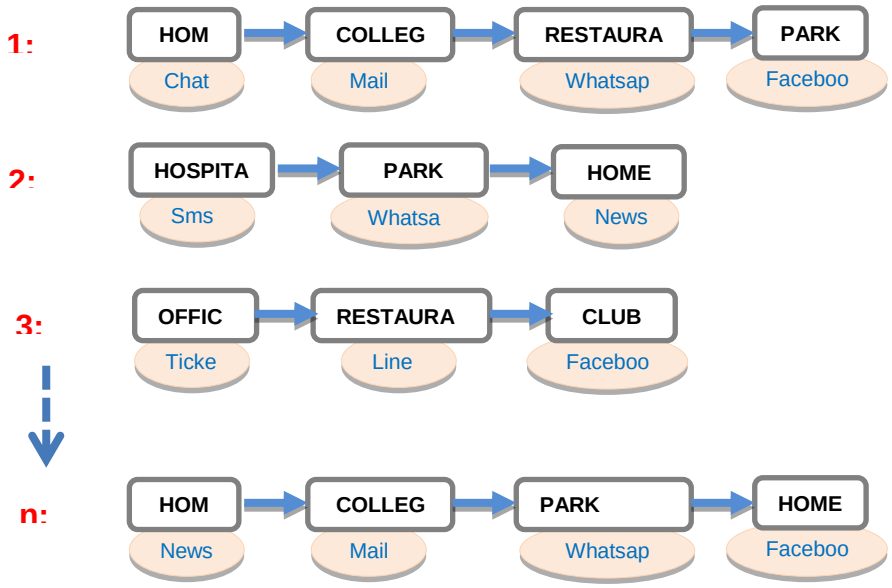



Figure 1: Mobile web service sequence generation scenario

In figure 1, different mobile web services are accessed by the mobile users at different locations. Here Home, Hospital, Restaurant, Park, etc., are the different locations and WhatsApp, News, Facebook, Chat, etc., are mobile web services which as denoted by $S_1$, $S_2$, $S_3$ and so on. Id 1, 2, 3... n represent different service access sequences, such as Id 1= {$S_5$, $S_3$, $S_1$, $S_6$}, Id 2= {$S_2$, $S_1$, $S_3$}, and so on [16], [17]. The traditional sequence pattern mining approach only considers the ítems that are sequentially purchased. They do not include any constraint or factor like price, profit or preferences of items. Sometimes the low frequency of items may be important. For example, let us assume there exists a pattern <gmail, facebook> in a sequence and also assume it is a low-frequency pattern in the sequence database. To handle this, Yun et al. proposed a new research approach, namely weighted sequential pattern mining [29], in which different weights are assigned to items by the importance of each item. Weighted frequent pattern mining [31] considers the importance of items and share frequent pattern mining [7] represents the occurrence of items in transactions as nonprofit binary values. In these frameworks, patterns with high weight value can be extracted even if they occur infrequently. Lan et al. [13] proposed an approach for finding weighted sequential patterns. They apply this approach to the traditional transaction and items. Some items with high profit but low count may not be discovered in a sequence database by using traditional sequential pattern mining approaches. To address this problem, Ahmed et al., proposed a new research approach, namely utility sequential pattern mining, which considers not only quantities and timestamp of items in sequence, but also individual profits of items in a quantitative sequence database [5].To find the most valuable pattern, utility plays an important role in data mining. Utility mining [23] has emerged as one of the most valuable research topics in the frequent pattern mining field. In utility mining, each item has an external utility such as profit, price and internal utility which indicates the non-binary value of items in transactional sequence [32].

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

In literature, various approaches are available for extracting useful frequent patterns using a utility from transactional databases. No current research addresses utility-based frequent pattern extraction form mobile web services sequences. Hence problem statement can be express as: *How to discovered frequent pattern from mobile web service accessing sequence using specific utilities?*

The utility-based approach can be used to discover frequent patterns from mobile web services sequences. Mobile web service frequent pattern mining is a new application of frequent pattern mining. In this regard, mobile web service accessed preference can be used as a utility value for a mobile web service. A utility-based approach is proposed to reduce a large number of candidate generations. Major contributions of this proposed work are summarized as follows:

1. The work uses an effectual sequence maximum utility (SMU) approach for the strong upper bound of utility support in subsequences.

2. The UBFPM (Utility Based Frequent Pattern Mining) approach has been proposed for finding interesting mobile web services patterns.

3. The proposed approach speeds up the execution efficiency in finding utility based patterns.

4. In the experiments, various datasets, both real and synthetic, are used to evaluate the performance of the proposed approach with state-of-the-art algorithms. The results show that the proposed UBFPM approach has good performance regarding execution time, memory consumption and a number of candidate generations.

The roadmap of the paper is as follows: the next section briefly recalls the history of the work related to this study. Preliminaries and problems are defined in section 3. Section 4 introduces our approach. Section 5 presents the experimental results on multiple datasets. The last section presents the conclusions.

# 2 Definitions and Related Work

To clearly describe frequent pattern mining approach for mobile web service, a set of relevant terms and related study is discussed in this section. It includes problem descrption and relevant definitions.

## 2.1 Description of the Problem and Definitions

Let us assume mobile web services with accessing sequence database given in Table 1, in which each row consists of two features, ID, and services sequence. There are six mobile web services in the dataset, respectively denoted as $S_1$ to $S_6$. We also assume the utility value of each mobile web service as shown in Table 2. Utility is a measure of how useful i.e. profitable a service is.Here accessing preference of a service is considered as an utility value. These values are randomly generated for experiment and assumed in the example.

Table 1: Service sequence dataset

| ID | Service sequence |
|---|---|
| 1 | $<S_1S_2S_3S_5>$ |
| 2 | $<S_2S_4S_5S_6>$ |
| 3 | $<S_4S_5S_6>$ |
| 4 | $<S_2S_3>$ |
| 5 | $<S_4S_6S_2>$ |
| 6 | $<S_1S_2><S_4S_6>$ |
| 7 | $<S_1S_2S_3><S_3S_6>$ |
| 8 | $<S_4S_2S_5>$ |
| 9 | $<S_1S_5S_6>$ |
| 10 | $<S_3S_4S_5S_6>$ |

Table 2: Service utility

| $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ |
|---|---|---|---|---|---|
| 0.9 | 0.4 | 0.2 | 0.8 | 0.5 | 0.6 |

We adopt definitions *similar* to those presented in the previous works [5], [6], [11], [16], [20], [22], [23], [31]. Let a set of web service *I* be {*S*1, *S*2, … *Sm*}. An itemset *X,* containing *k* items, is called *k-itemset* and its length is *k*. A

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

sequence is an ordered list of services, such as $ID_8$=<$S_4S_2S_5$>. A mobile web service sequence database *D*= {*ID*1, *ID2*, … *IDn*} [28] contains *n* sequence itemsets which are a subset of itemset *I*.

Definition 1, Sub and super sequences: Let A and B be two sequences. A=<$X_1$ $X_2$ $X_3$...$X_m$> and B=<$Y_1$ $Y_2$ $Y_3$...$Y_n$>. If there exist an integer 1≤ $i_1$< $i_2$<... $i_m$≤ n such that $X_1 \subseteq Y_{i1}$,$X_2 \subseteq Y_{i2}$ ,….,$X_m \subseteq Y_{im}$ , then A is called a subsequence of B, and B is called a super sequence of A [11].

Definition 2, Utility value: The utility value of a mobile web service *S*, ranges from 0 to 1. The utility is associated with each service; it indicates the accessing preference value of the service [11].

Definition 3, Utility of service set: Let X = ($S_1$ $S_2$ … $S_m$). The utility of the mobile web services set X is the summation of utility values of all mobile web services which belong to X, divided by the cardinality of X [11].

$$Ux = \frac{\sum_{i=1}^{|X|} Ux(i)}{|X|} \tag{1}$$

$U_{x(i)}$is the utility value of the mobile web service that appears in the position i of X.

For example, let X= ($S_1$ $S_3$ $S_5$), based on the utility values presented in the table 2 is 0.9, 0.2, 0.5 respectively and |X|=3 therefore:

$$Ux = \frac{0.9 + 0.2 + 0.5}{3} = 0.54 \tag{2}$$

Definition 4, Utility of a sequence: Let p= ($X_1$ $X_2$ $X_3$ … $X_n$) The utility value of the service sequence $ID_2$ ($U_p$) is the summation of the utility values of all mobile web services sets which belong to p, divided by the cardinality of p [11].

$$Up = \frac{\sum_{i=1}^{|p|} Ux(i)}{|p|} \tag{3}$$

$U_{x(i)}$ is the utility value of the mobile web services set $X_i$ that appears in the position i of p.

For example, in Table 2, since the utility values of the sequence <$S_4S_6S_2$> are 0.8, 0.6 and 0.4 respectively, and the number of mobile web service in sequence <$S_4S_6S_2$> is 3, then U<$S_4S_6S_2$>=(0.8+0.6+0.4)/3=0.6.

Definition 5, Utility of a subsequence: The utility value of a mobile web services subsequence *r*, $S_r$ is the summation of utility values of all mobile web services in *r* over the number of sequence in *r* [11]. That is,

$$Sr = \frac{\sum_{x \in ID}^{ID} Ux}{|r|} \tag{4}$$

Where |*r*| and $U_x$ are the number of mobile web services in the subsequence *r,* and the utility value of mobile web services set *S* in *r*, respectively.

For example, in Tables 1 and 2, since the sixth sequence <$S_1S_2$><$S_4S_6$> consists of two sequences {$S_1S_2$} and {$S_4S_6$}, and the utility of two sequences are 0.65 and 0.7 respectively, then *U*<$S_1S_2$><$S_4S_6$> =(0.65+0.7)/2=0.675.

Definition 6, Sequence max utility and Total Sequence max utility (TSMU): The sequence max utility value of a sequence *p*, *SMU*, is the maximum utility value among all mobile web services in the sequence *p* [11]. The total *SMU* of database *D, TSMU*, is the summation of the *SMU* values of all sequences in *D* [11]. That is,

$$TSMU = \sum_{i=1}^{|D|} SMU_{p(i)} \tag{5}$$

Where |D| represents the cardinality of the mobile web services accessing database (D).

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

For example, in Table 1, the *SMU* of ten sequences are 0.9, 0.8, 0.8, 0.4, 0.8, 0.9, 0.9, 0.8, 0.9 and 0.8 respectively. Then *TSMU*=0.9+0.8+0.8+0.4+0.8+0.9+0.9+0.8+0.9+0.8=8.0.

Definition 7, Utility support value: Utility support value (USV) in a sequence calculated as the addition of the sequence utility values [11]. It is given as

$$USV_p = \frac{U_{p*t}}{TSMU} \tag{6}$$

Where *t* is the number of times where *p* appears as a subsequence in all the sequences of the database D.

For example, if we want to find the USV of sequence ID 4, i.e., $<S_2S_3>$ then it is

$$USV_{p4} = \frac{U_{p4*t}}{TSMU} = \frac{0.3*3}{8} \approx 11\% \tag{7}$$

(*t=3* because $p_4$ is a subsequence of ID 1, 4 and 7, as of definition 1).

Definition 8, Utility frequent sequential pattern: A subsequence *r* is called a utility frequent sequential pattern *(FSP)* if *USV* ≥ *min_uti,* where *min_uti* is a predefined minimum utility threshold.

For example in Table 1, a $S_2$ service appears 7 times, and its utility is 0.4 in Table 2. Then its *USV* is (0.4+0.4+0.4+0.4+0.4+0.4+0.4)/8=35%. If *min_uti*=25%, then the sequence $<S_2>$ is a FSP.

Definition 9, Utility sequence upper bound**:** The utility sequence upper bound pattern of a subsequence *r*, is the sum of *SMU* of the sequence including *r* in sequence database over the *TSMU* of the *D* [11]. It is denoted as *SUB* and is defined as

$$SUB_r = \left\{ \frac{\sum_{i=1}^{|D|} SMU_{r(i)}}{TSMU} \right\} | \, r \text{ is subsequence of } \mathrm{r(i)} \wedge \mathrm{r(i)} \in \mathrm{D} \tag{8}$$

For example, in Table 1, the sequence $<S_5>$is a subsequence of ID 1, 2, 3, 8, 9 and 10. Therefore,

$$SUB_{<S5>} = \frac{SMU(1) + SMU(2) + SMU(3) + SMU(8) + SMU(9) + SMU(10)}{TSMU}$$
$$SUB_{<S5>} = \frac{0.9 + 0.8 + 0.8 + 0.8 + 0.9 + 0.8}{8} \approx 62\% \tag{9}$$

Definition 10, Utility frequent sequence upper bound pattern: A subsequence *r* is called utility frequent sequence upper bound pattern *(FSUBP)* if *SUB*≥ *min_uti.*

Problem statement: Mobile web service frequent pattern mining is a new application of frequent pattern mining as well as mobile computing. If a particular user visited multiple locations on a full day (24 hours), we have a sequence dataset about their visiting as well as service details which is linked to the user. Based on this information we can extract some new knowledge and facts. Utility based approach can be used to discovered frequent patterns from mobile web services sequences. In this regard, mobile web service accessed preference can be used as a utility value for a mobile web service. Suppose a user accessed a particular service 20 times in a complete day. This can be expressed as 0.2 utility values. To address the above reason, we propose a utility based approach to reduce the large number of generated candidates. The problem is to find a complete set of frequent service patterns in database *D*.

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

## 2.2 Related Work

In this subsection, related work on frequent pattern mining, utility mining, utility-based frequent pattern mining and sequential pattern mining is briefly reviewed.

### 2.2.1 Frequent Pattern Mining

The process of extracting a set of items or subsequences that occur frequently in a dataset is known as frequent pattern mining. Different studies have been conducted to mine frequent patterns through transactional databases. Firstly, Apriori started mining frequent itemsets from transactional databases [1]. To get a better result than Apriori, FP-Growth method was later develop [10]. FP-Growth has only one scan of the database. Hence it improved the efficiency of the algorithm. Several types of databases such as sequential, incremental, and stream are used for frequent pattern mining [31]. The relative importance of items in databases can be found through weighted frequent pattern mining [32]. In this method weight of a pattern is calculated by dividing the sum of weights of items by pattern length.

### 2.2.2 Utility Mining

In the transactional database, a profit, weight, importance or performance of an item can be considered as utility value [13], [17]. The utility was firstly used by Chan et al. in transactional databases [8]. To prune the search space, an estimation method was used named as Umining [24]. Level-wise search method uses item discarding approach to reduce candidate generation [14]. Various required information on utility mining is maintained using a tree-structured, known as Huc-Tree [6]. To discover high utility itemset, it is required to maintain downward closure property. It is done by transaction weighted utilization model, which is based on Apriori algorithm [15]. Apriori-based utility mining approaches use multiple database scans for candidate generation. Another method, Incremental High Utility Pattern (IHUP) was proposed by Ahmed et al. to avoid multiple database scans that uses FP-Tree concept [5]. To enhance the performance of utility mining and getting higher itemset Tseng et al. proposed UP-Growth [22] algorithm, which included various strategies for mining. Next revised version of UP-Growth is UP-Growth+ [23], it decreases overestimated utilities. A tree-based high utility itemset mining algorithm MU-Growth is proposed by Yun et al. [31] which reduces the number of candidates.

### 2.2.3 Utility Based Frequent Pattern Extraction

High utility items may occur with a low frequency but have more importance. In a transaction, the gold item may have low frequency, but its value is higher. Transactional association rule mining [1] approaches use 0 or 1 for ítem absent or present in a sequence. Traditional approaches are not sufficient for high utility with low frequencies, then, utility-based approaches are useful for frequent pattern extraction. To fulfill a business objective, Chan et al. proposed the idea of top-K patterns [8]. To discover valuable frequent itemset, weighted itemset mining has been proposed [28]. Yun et al. also uses an upper bound model to handle downward closure property [28]. Later, to enhance the performance of weighted itemset mining various studies have been proposed [3], [4], [5], [9], [26], [27].

### 2.2.4 Utility Based Sequential Pattern Extraction

Transactions and timestamps are present in a sequential dataset. This dataset consists of the transaction Id, consumer detail and list of buying items. Frequent pattern generation is possible using these datasets. Agrawal et al. proposed the AprioriAll, AprioriSome and DynamicSome algorithms for sequential pattern mining [2]. Generalized Sequential Patterns (GSP) [21] and PrefixSpan [19] approaches were later developed for enhancing execution efficiency in sequential pattern mining. Yun et al. proposed various approaches to find weighted sequential patterns in sequential databases [30]. Shie et al. proposed a valuable pattern mining approach to discover high utility itemsets in different shopping websites using quantities and profits [20]. Next, Ahmed et al. proposed a new research approach, high utility sequential pattern mining, in which they consider the relationship order of an itemset with quantity and profit. According to traditional sequential pattern mining, the count of a pattern in a sequence was only regarded as one even if the subsequence appeared multiple times in a sequence. Based on this concept, max utility concept could be more suitable regarded as the estimated utility for subsequence in quantitative sequences [5, 11]. Consumers' purchase behavior extraction is the main use of sequential pattern mining. Max utility concept may be more appropriate for finding high utility sequential patterns in various real life problems, such as getting high utility business policies or finding mostly accessed services based on their preferences [11]. Lan et al. proposed a sequence utility upper bound model for generating patterns [12]. This model did not adopt any strategy to handle the high utility sequential pattern mining task. Lots of unpromising subsequences still need to be generated. In addition, the USpan approach has to spend a great deal of execution time using an LQS-Tree structure [25]. Thus, the utility upper bound reduction for subsequences in mining is quite important. The aim of this study is to develop an efficient approach to extract frequent patterns from mobile web service sequences.

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

# 3 Proposed Approach

The proposed approach effectively handles the problem of finding utility frequent sequential patterns from mobile web service sequences. We have adopted minimum utility value to mine frequent utility patterns. The proposed approach, UBFPM, can reduce the number of candidates for utility itemsets and reveals valuable information that may be needed in various applications of user behavior analysis.

The proposed approach mainly consists of four steps: first it finds FSP-1 and FSUBP-1 patterns. Second, sequences are modified and new SMU's are calculated based on FSUBP-1 patterns. Next, a postfix database is generated for each frequent web service. Last, this process recursively applies for generating FSP-n patterns using postfix databases. Section 3.1 describes the generation of FSP-1 and FSUBP-1 patterns; Section 3.2 describes sequence modification and calculation of new SMU's; Section 3.3 describes the process of postfix database preparation; Section 3.4 describes the process of FSP-n patterns generation.

## 3.1 FSP-1 and FSUBP-1

In this step, sequence database is scanned, and the SMU value of each sequence is calculated. It is shown in Table 3.

Table 3: SMU values of sequence

| ID | Service sequence | SMU |
|---|---|---|
| 1 | $<S_1S_2S_3S_5>$ | 0.9 |
| 2 | $<S_2S_4S_5S_6>$ | 0.8 |
| 3 | $<S_4S_5S_6>$ | 0.8 |
| 4 | $<S_2S_3>$ | 0.4 |
| 5 | $<S_4S_6S_2>$ | 0.8 |
| 6 | $<S_1S_2><S_4S_6>$ | 0.9 |
| 7 | $<S_1S_2S_3><S_3S_6>$ | 0.9 |
| 8 | $<S_4S_2S_5>$ | 0.8 |
| 9 | $<S_1S_5S_6>$ | 0.9 |
| 10 | $<S_3S_4S_5S_6>$ | 0.8 |
| **TSMU value** | | **8.0** |

After getting SMU and TSMU, we generate Subsequence-1 SUB and USV values of all the service which is shown in Table 4.

Table 4: Subsequence-1 SUB and USV

| **Subsequence** | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ |
|---|---|---|---|---|---|---|
| **SUB** | 45% | 69% | 37% | 61% | 62% | 74% |
| **USV** | 45% | 35% | 10% | 60% | 37% | 52% |

After applying minimum utility threshold *min_util*=50%, following FSP-1 and FSUBP-1 patterns are discovered.

*FSUBP-1* patterns: $S_2$, $S_4$, $S_5$ and $S_6$.
*FSP-1* patterns: $S_4$, and $S_4$.

## 3.2 Modified SMU and Sequence Generation

Four services, $S_2$, $S_4$, $S_5$ and $S_6$ are above the minimum utility threshold. These services can be used to generate the next level sequential patterns. They are also used to modify sequences and SMU values. These modified values are shown in Table 5.

Table 5: Modified SMU and sequences

| ID | Service sequence | SMU |
|---|---|---|
| 1 | $<S_2S_5>$ | 0.5 |
| 2 | $<S_2S_4S_5S_6>$ | 0.8 |
| 3 | $<S_4S_5S_6>$ | 0.8 |
| 4 | $<S_2>$ | 0.4 |
| 5 | $<S_4S_6S_2>$ | 0.8 |
| 6 | $<S_2><S_4S_6>$ | 0.8 |

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

| *Table 5: continuation* | | |
|---|---|---|
| 7 | <$S_2$><$S_6$> | 0.6 |
| 8 | <$S_4S_2S_5$> | 0.8 |
| 9 | <$S_5S_6$> | 0.6 |
| 10 | <$S_4S_5S_6$> | 0.8 |

## 3.3 Postfix Sequence Generation

According to *FSUBP-1* patterns, postfix sequences are generated. In postfix sequence, only services which follow *FSUBP-1* patterns are considered. Below Table 6 shows the postfix sequence of *FSUBP-1* pattern <$S_4$>.

Table 6: Postfix sequence of *FSUBP-1* pattern <$S_4$>

| ID | Service sequence | SMU |
|---|---|---|
| 1 | <$S_4S_5S_6$> | 0.8 |
| 2 | <$S_4S_5S_6$> | 0.8 |
| 3 | <$S_4S_6S_2$> | 0.8 |
| 4 | <$S_4S_6$> | 0.8 |
| 5 | <$S_4S_2S_5$> | 0.8 |
| 6 | <$S_4S_5S_6$> | 0.8 |

## 3.4 Recursively Generating FSP-n and FSUBP-n

The above steps are recursively applied to generate FSP-n and FSUBP-n patterns. Based on the *FSUBP-1* pattern $S_4$, the following subsequence-2 can be generated.

*FSUBP-2* patterns: $S_4$ $S_6$.
*FSP-2* patterns: Nil.

In a similar way, all the FSP-n and FSUBP-n patterns can be generated. The algorithm UBFPM for generating utility frequent sequential patterns is shown below.

**Algorithm 1: Algorithm of generating utility frequent sequential patterns**
**Input:** Web service sequence database D, utility values and *min_util* threshold
**Output:** FSP-n and FSUBP-n patterns

```
1:  Scan database D and compute SMU,TSMU
2:  for each Si ∈ D do
3:   {
4:   Calculate SUBi and USVi
5:                    if(SUBi ≥ min_util)
6:                   {
7:                   Generate FSUBP-1 patterns
8:                   }
9:                    if(USVi ≥ min_util)
10:                  {
11:                  Generate FSP-1 patterns
12:                   }
13:  }
14:  for each IDi ∈ D do
15:  {
16:                  if(Si∉SUB1)
17:                  {
18:                   Modify SMU and sequences
19:                  }
20:  }
21:  for each SUBi ∈ SUB-1 do
22:  {
23:                  for each Si ∈ SUB-1 do
24:                  {
25:                  Generate postfix sequence Si
26:                  }
27:  }
28:  for each postfix sequence Si do
29:  {
30:       Calculate SUBn and USVn of Si
```

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

31: if($USV_n$ subsequence of $S_i \geq$ *min_util*)
32: {
33: Generate FSP-n patterns
34: }
35: }
36: Print all FSP-n patterns

# 4 Experiments

In this section, we evaluate the performance of our algorithm UBFPM. The experiment was performed on a Pentium Dual-Core 3.3 GHz processor with 8 GB of memory, using the Java programming language. The experiments ran in the Windows 7 operating system. The simulation is performed on both a synthetic and a real database. The performance of the proposed UBFPM approach is compared with state-of-the-art pattern mining approaches such as IHUP [5],Up-growth [22], UP-Growth+ [23] and MU-Growth [31].

## 4.1 Experiment on Synthesis Dataset

In this experiment, we use the public IBM data generator (Site 2). This data generator produces the mobile web services sequence data. The parameters used in the IBM data generator were the average length of transaction per sequence S, the average length of services per transaction T, the average length of maximum potentially frequent services set I, the total number of distinct mobile web services N, and the total number of sequences D. For each service sequence dataset generated, a corresponding utility table was also produced in which a utility value in the range from 0.0 to 1.0 was randomly assigned to a service. The simulation model was similar to that used in Liu et al. [15], to generate the utilities of the services in the sequence. Figure 2 shows the utility-value distribution of all the mobile web services generated by the simulation model in the utility table.

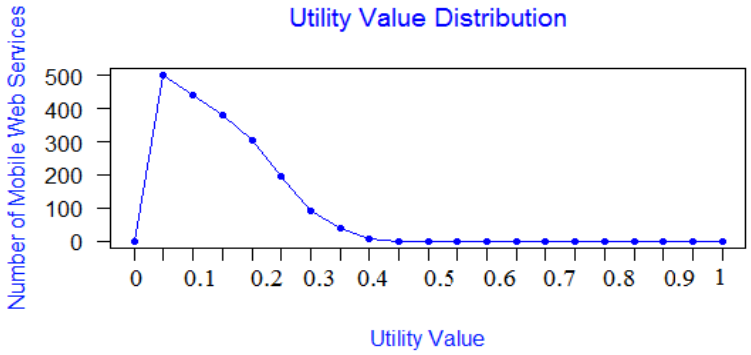


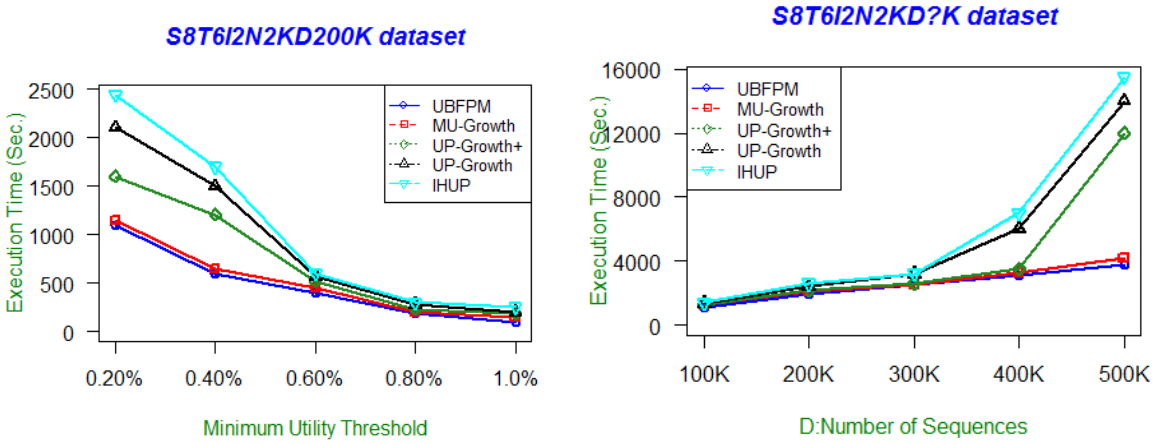

Figure 2: The utility-value distribution in the synthetic dataset

## 4.2 Performance Comparison on the Synthetic Dataset

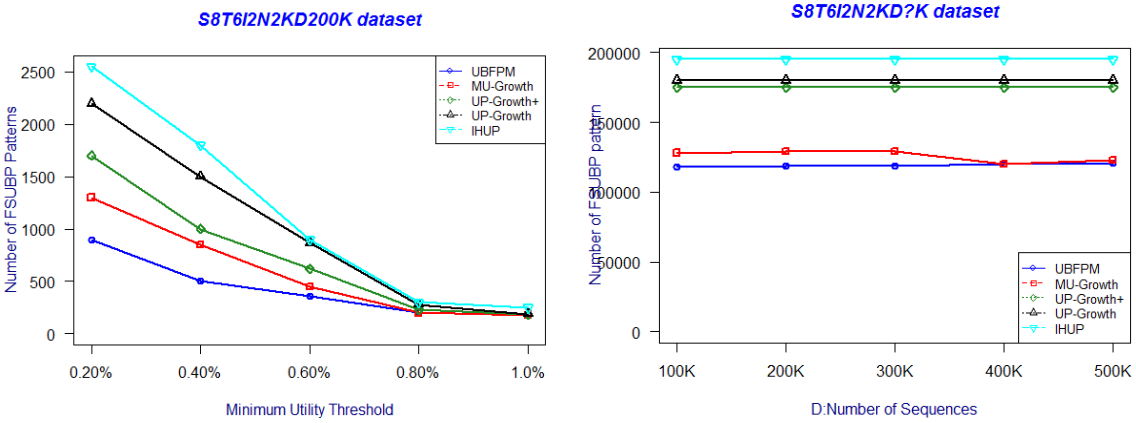
Figure 4 shows the experimental results of performance evaluation on a synthetic dataset. Figures 3 (a) and (b) present the results of total execution time. Figures 3(c) and (d) present the number of *FSUBP* patterns on fixed data size (200k) and varied data set size, respectively. In figure 3, the proposed approach UBFPM has the best performance in terms of total execution time as well as the lowest memory consumption. Other approaches generate a more *FSUBP* pattern, while UBFPM generates less frequent patterns. In figures 3 (a) and (b) the approach takes less time as compared to different state-of-the-art approaches because these use tree-based pruning strategy. Tree-based pruning requires more time to construct the tree first, and then prunes based on a minimum utility threshold. In terms of execution time, the approach is more efficient while the minimum utility threshold is less than 0.60%. As seen in figure 3 (a), when the minimum utility threshold increases from 0.20% to 0.60%, execution time is varied for all approaches. But when the minimum utility is higher than 0.60% this variation goes down and above 1% execution time is approximately similar. The same thing is applicable to different data sizes. As figure 3 (b) shows for small data size (100k) all algorithms take approximately the same execution time, but when the number of sequences is increased (about 200k or more), the previous algorithm takes more time, while UBFPM performs well.

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at www.jtaer.com
DOI: 10.4067/S0718-18762019000300106




(a) Execution Time on S8T6I2N2KD200K (b) Execution Time on S8T6I2N2KD?K




(c) Number of *FSUBP* patterns (d) Number of *FSUBP* patterns

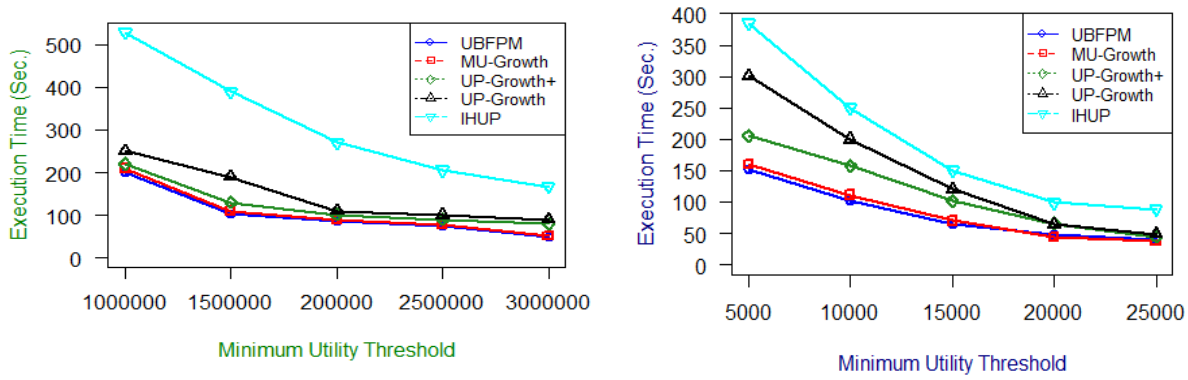
Figure 3: Performance comparison on the synthetic dataset

## 4.3 Performance Comparison on Real Datasets

We present the experimental results of the compared approaches under varied minimum utility values in figure 4. For this performance comparison, the kosarak and retail dataset are used. For both datasets different minimum utility thresholds have been used. In figure 4 (a), the runtime of the UBFPM is best among all other approaches on the kosarak dataset. The results show that the proposed approach is more suitable while minimum utility threshold is increasing from smaller to higher. In addition, it is observed that the approach generates the least number of frequent *FSUBP* patterns. Another comparison is shown in figure 4 (b) on the retail dataset. When minimum utility threshold is increased from 5000 to 20000, the execution time decreases. In this figure, it is shown that all the approaches have a good execution time beyond the 20000 minimum utility thresholds. Figure 4 shows that MU-Growth also performs well for both datasets.

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 14 / ISSUE 3 / SEPTEMBER 2019 / 76-88

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

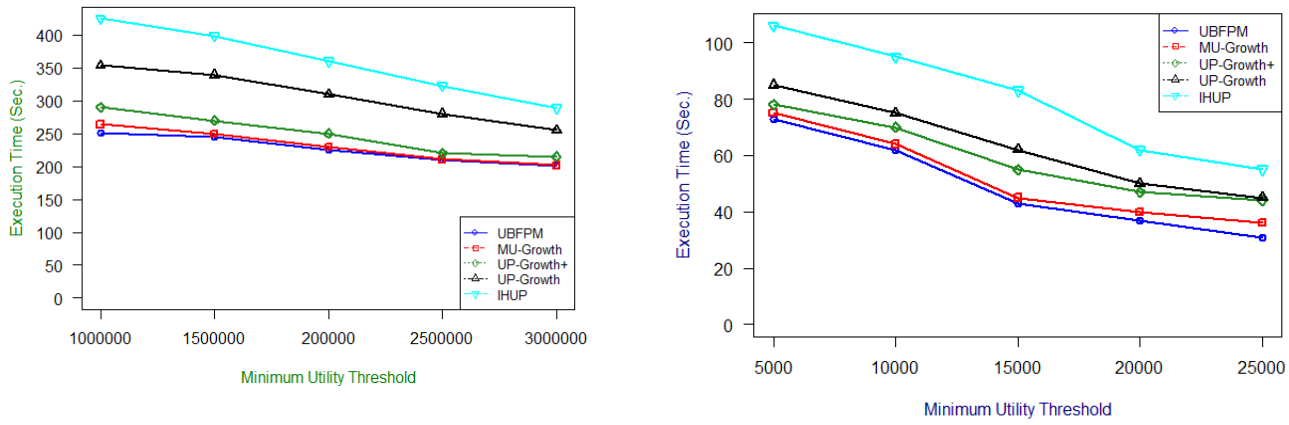



(a) Running time on kosarak dataset (b) Running time on retail dataset

Figure 4: Execution time comparison on real datasets

## 4.4 Memory Usage

Figure 5 shows the memory usage of the different approaches to different datasets. UBFPM always usage less memory than the other algorithms. The reason is that these algorithms have to reserve a very large amount of memory to store candidate itemsets during the execution process, while UBFPM does not. Figures 5 (a) and (b) show memory consumption on the kosarak and retail dataset, respectively. Figure 5 (a) indicates that when the minimum utility increases from 1000000 to 3000000, memory usage gradually decreases. In this figure the rate of memory usage also decreases for other approaches, but UBFPM frees more memory space for execution. Memory usage is also shown in figure 5 (b) for the retail dataset. When minimum utility increases from 5000 to 25000, memory usage decreases.




(a) Memory usage on kosarak dataset (b) Memory usage on retail dataset

Figure 5: Memory usage comparison on real datasets

## 4.5 Discussion

Having run the above experiments, the proposed approach UBFPM is shown to outperform the current state-of-the-art algorithms. To mine interesting patterns, almost all existing algorithms first generate candidate itemsets and subsequently compute the exact utility of each candidate to identify interesting patterns. The UBFPM approach does not generate candidate sets as it stores only postfix sequences of services. Experimental results showed that UBFPM extracts frequent patterns faster than the state-of-the-art approaches [5], [22], [23], [31]. Figures 3, 4 and 5 show that the UBFPM approach reduces the execution time as well as memory usage. The number of frequent patterns generated by the proposed approach was clearly less than that of existing algorithms. The main reason for this is that the maximum utility value in a services sequence was more suitable as an upper bound of any subsequence in a sequence. Today mobile devices and services are getting more and more popular for various applications. Effective data mining techniques for ensuring user's requirements in both of reliability and timeliness on the mobile devices with limited resources is still a crucial challenge. Regarding reliability, our proposed approach is

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

shown to produce highly frequent patterns as illustrated by the rationale in section 4. Hence, the proposed approach does not only meet timeliness requirements but also identifies interesting patterns for users.

# 5 Conclusion

In this paper, we propose an efficient approach named UBFPM, for service frequent pattern extraction using utility as the preference of service. We also propose an algorithm which is based on the postfix sequence generation of service sequence. More accurate frequent upper bounds are also computed for enhancing the filtration of service sequence. The proposed approach can discover highly frequent *FSUBP* patterns and sequential *FSP* patterns of service sequences. These discovered patterns are very useful for mobile web service users and business analysts. If a service provider knows the frequent patterns of any sequence beforehand, they can take decisions to enhance their business effortlessly. The experimental results show that UBFPM is better than previously developed approaches. With the help of this approach, mobile web services prediction and maintenance becomes simpler and easier. In the next steps, we will attempt to handle the dynamic maintenance problem of utility based sequential patterns, when sequences are dynamically modified.

# Acknowledgments

The author would like to thank the Central University of Rajasthan for providing a workplace, support and resources.

# Websites List

Site 1: Frequent Itemset Mining Dataset Repository
http://fimi.ua.ac.be/data/

Site 2: IBM Quest Data Mining Project, Quest Synthetic Data Generation Code, Available at
http://www.almaden.ibm.com/cs/quest/%20syndata.html

# References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules, in Proceedings 20th International Conference Very Large Data Bases, VLDB. San Francisco, CA. 1994, pp. 487-499.
[2] R. Agrawal and R. Srikant, Mining sequential patterns, in Proceedings of the Eleventh International Conference on Data Engineering. IEEE, Taipei, Taiwan, 1995.
[3] C. F. Ahmed, et al. Single-pass incremental and interactive mining for weighted frequent patterns, Expert Systems with Applications, vol. 39, no. 9, pp. 7976-7994, 2012.
[4] C. F. Ahmed, et al. Handling dynamic weights in weighted frequent pattern mining, IEICE TRANSACTIONS on Information and Systems, vol. 91, no.11, pp. 2578-2588, 2008.
[5] C. F. Ahmed, et al. Efficient tree structures for high utility pattern mining in incremental databases, IEEE Transactions on Knowledge and Data Engineering, vol. 21, no.12, pp. 1708-1721, 2009.
[6] C. F. Ahmed, et al. HUC-Prune: an efficient candidate pruning technique to mine high utility patterns, Applied Intelligence, vol. 34, no. 2, pp. 181-198, 2011.
[7] B. Barber and J. H. Howard Extracting share frequent itemsets with infrequent subsets, Data Mining and Knowledge Discovery, vol. 7, no. 2, pp. 153-185, 2003.
[8] R. Chan, Y. Qiang and S. Y. Dong, Mining high utility itemsets, in Proceedings Third IEEE International Conference on Data Mining. IEEE, Melborune, FL, 2003, p. 9.
[9] J. H. Chang, Mining weighted sequential patterns in a sequence database with a time-interval weight, Knowledge-Based Systems, vol. 24, no. 1, pp. 1-9, 2011.
[10] J. Han, J. Pei and Y. Yin, Mining frequent patterns without candidate generation, ACM Sigmod Record, vol. 29, no. 2, 2000.
[11] G. C. Lan, T. P. Hong and H. L.Lee, An efficient approach for finding weighted sequential patterns from sequence databases, Applied Intelligence, vol. 41, no. 2, pp. 439-452, 2014.
[12] G. C. Lan, T. P. Hong and V. S. Tseng, Sequential utility mining with the maximum measure, in Proceedings of The 29th Workshop on Combinatorial Mathematics and Computation Theory, Taipei, Taiwan, 2012, pp. 115-119.
[13] G. C. Lan, T. P. Hong and V. S. Tseng, and S. L. Wang, Applying the maximum utility measure in high utility sequential pattern mining, Expert Systems with Applications, vol. 41, no.11, pp. 5071-5081, 2014.
[14] Y._C. Li, J.-S. Yeh and C.-C.Chang, Isolated items discarding strategy for discovering high utility itemsets, Data & Knowledge Engineering, vol. 64, no. 1, pp. 198-217, 2008.
[15] Y. Liu, W. K. Liao and A. N. Choudhary, A two-phase algorithm for fast discovery of high utility itemsets, in Proceedings Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, 2005, pp. 689-695.
[16] K. K. Mohbey, High fuzzy utility based frequent patterns mining approach for mobile web services sequences, International Journal of Engineering-Transactions B: Applications, vol. 30, no. 2, pp. 182-191, 2017.

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762019000300106

[17] K. K. Mohbey, Utility based frequent pattern extraction from mobile web services sequence, Journal of Information Technology Research (JITR), vol. 11, no. 2, pp. 31-52., 2018
[18] K. K. Mohbey and G. S. Thakur, Constraint based interesting location and mobile web service sequence mining in M-commerce environment, Journal of Theoretical and Applied Electronic Commerce Research, vol. 11, no. 1, pp. 84-95, 2016.
[19] J. Pei, J. Han, B.M. Asi, J. Wang, and Q. Chen. Mining sequential patterns by pattern-growth: The prefixspan approach, IEEE Transactions on Knowledge & Data Engineering, vol. 11, pp. 1424-1440, 2004.
[20] B. E. Shie, P. S. Yu and V. S. Tseng, Mining interesting user behavior patterns in mobile commerce environments, Applied Intelligence, vol. 38, no. 3, pp. 418-435, 2013.
[21] R. Srikant and R. Agrawal, Mining sequential patterns: Generalizations and performance improvements, in Proceedings International Conference on Extending Database Technology, Springer, Berlin, Heidelberg, 1996, pp. 1-17.
[22] V. S. Tseng, C. W. Wu, B. E. Shie, and P. S. Yu, UP-Growth: an efficient algorithm for high utility itemset mining, in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, Washington, DC, 2010, pp. 253-262.
[23] V. S. Tseng, B.-E. Shie, C.-W. Wu, and P. S. Yu, Efficient algorithms for mining high utility itemsets from transactional databases, IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 8, pp. 1772-1786, 2013.
[24] H. Yao and H. J. Hamilton, Mining itemset utilities from transaction databases, Data & Knowledge Engineering, vol. 59, no. 3, pp. 603-626, 2006.
[25] J. Yin, Z. Zheng and L. Cao, USpan: An efficient algorithm for mining high utility sequential patterns, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, Beijing, China, 2012, pp. 660-668.
[26] U. Yun, A new framework for detecting weighted sequential patterns in large sequence databases, Knowledge-Based Systems, vol. 21, no. 2, pp. 110-122, 2008.
[27] U.Yun, On pushing weight constraints deeply into frequent itemset mining, Intelligent Data Analysis, vol. 13, no. 2, pp. 359-383, 2009.
[28] U. Yun and J. J. Leggett, WFIM: Weighted frequent itemset mining with a weight range and a minimum weight, in Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.
[29] U. Yun and J. J. Leggett, WSpan: Weighted sequential pattern mining in large sequence databases, in Proceedings 3rd International IEEE Conference Intelligent Systems, London, UK, 2006, pp. 512-517.
[30] U. Yun, J. J. Leggett and. T. J Ong, Mining weighted sequential patterns based on length-decreasing support constraints, in Proceedings International Conference on Intelligence and Security Informatics, Springer, Berlin, Heidelberg, 2006, pp. 650-651.
[31] U. Yun, H. Ryang and K.H. Ryu, High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates, Expert Systems with Applications, vol. 41, no. 8, pp. 3861-3878, 2014.
[32] U. Yun, G. Lee and K. Ryu, Mining maximal frequent patterns by considering weight conditions over data streams, Knowledge-Based Systems, vol. 55, pp. 49-65, 2014.