An improved Frequent Pattern Mining in Sustainable Learning Practice using Generalized Association Rules

R. Revathy, S. Balamurali

Abstract: ***This research focuses on mining the frequent patterns occurred in the given Datasets by Generalization of Association Rules. Frequent pattern mining is a significant as well as interesting problem in the research filed of Data Mining. Building of frequent pattern tree (FP tree), frequent pattern growth (FP growth) and association rule generation are implemented to find the repeated patterns of data. FP tree Construction Algorithm is mainly used to compact a vast database into a extremely compressed, seems to very tiny data structure; hence eliminates the repeated scanning of database. The role of FP growth algorithm is to discover the frequent patterns with FP tree structure and construct the generalized association rules corresponding to the learning data. FP growth algorithm obtained best results as compared with conventional Apriori algorithm. This research provides some practical real time applications pertaining data mining techniques in the field of learning, education, marketing, finance and so on.***

Keywords: ***Data mining, Concept hierarchy, FP growth algorithm, Association rule.***

I. INTRODUCTION

The work mainly deals with information of multi level hierarchy and process of mining the frequent patterns are integrated by providing users with skilled learning demands and various learning proposals to learners. The main objective in this research is employing the FP growth that allows mining the entire set of repeated patterns with FP tree with the help of fragment growth patterns and subsequently, generates the association rules. Discovering patterns from huge databases and consuming the related information in different aspects is referred as data mining. It helps in various disciplines as given in Fig.1 for searching appropriate solutions of their problems [1].

In data mining, FP growth algorithm performs FP tree construction and achieves proficient mining on this tree. Association rule generation process searches for interesting relationship among the data and the rule is generated from the tree [2]. While constructing FP tree, frequent access rule will be produced from the combination of conditional frequent patterns. As a result, pruning technique is implemented to eliminate the items that occurred in the same group.

Revised Manuscript Received on December 16, 2019.

R. Revathy, Department of Computer Applications, Kalasalingam Academy of Research and Education, Krishnankoil, Tamil Nadu, India. Email: revathy.r@klu.ac.in.

S. Balamurali*, Department of Computer Applications, Kalasalingam Academy of Research and Education, Krishnankoil, Tamil Nadu, India. Email: sbmurali@rediffmail.com.

Statistics
Data visualisation
Data Mining
Database management
Artificial Intelligence

Fig. 1. Interdisciplinary structure of Data mining

The preliminary motivation of producing association rules replaces the drawbacks of market basket analysis [3]. Conventional system is partially based on the dynamic item counting method (DIC) and probably follows physical calculation scheme. It is somehow difficult to hit upon the frequent patterns from the bulky transaction using DIC method. The Apriori algorithm, a conventional method which follows a procedure called layer-by-layer iteration [4]. It uses extra database scans in a repeat manner for tracking itemsets in the large database. High requirement of the time and lacking in flexibility of user interaction are the main complication of processing the large transaction database [5].

To overcome the drawbacks of the traditional system, an efficient algorithm called FP growth which is used in this work to mine the frequent patterns from large itemset. FP growth is an algorithm for generating frequent item sets for association rules and compresses a large database into a compact FP tree structure [6]. This research implemented FP growth algorithm to determine the most frequent data from the learning dataset. When comparing with apriori algorithm, FP growth is faster in execution; transaction database doesn't require loading in every iteration.

II. MATERIAL AND METHODS

Frequent pattern mining is nothing but finds remarkable relationship i.e., association or correlation among huge set of data items. By way of immense quantity of data, the items are continuously collected and then stored. Hence, many fields are becoming interested in mining association rules of their databases to resolve the real time problems [7]. This paper holds four phases in implementing of learning database as follows. The work flow of the corresponding modules is depicted in Fig. 2.

- Filtering phase
- Tree Construction phase
- Mining Frequent Patterns
- Pruning phase

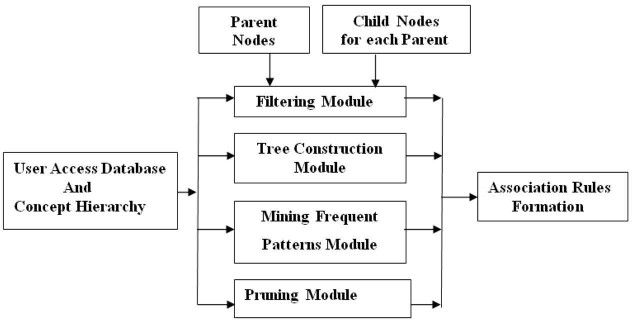

Parent Nodes
Child Nodes for each Parent
Filtering Module
Tree Construction Module
Mining Frequent Patterns Module
Pruning Module
User Access Database And Concept Hierarchy
Association Rules Formation

Fig. 2. Working process of modules

A. Filtering

In multiple level association rule mining, the itemset are represented by means of concept hierarchy. Process of mining happens at each level in the hierarchy. The top-down procedure is usually used in concept hierarchy where the value of support threshold fluctuates from one level to another level [8]. For example, the organization information of the learning curriculum system is hierarchically designed as shown in Fig. 3. It has a unique encoded string referred as accessing-identifier, which is enclosed by parenthesis for each identifier.

Table- I: User access database

User ID	Items set
Usr 1	{321, 231,312, 322}
Usr 2	{112, 321,212,111,322,231}
Usr 3	{321, 322, 212,111, 231}
Usr 4	{111,112, ,222,211,212}
Usr 5	{311,211, 112,212}
Usr 6	{111,311,212,112,231}
Usr 7	{231,121,321}

In this scheme, the learning system is constructed as three levels of hierarchical model. The first level includes the departments those come under the curriculum whereas the second level represents the courses belong to the departments. Finally, the third level enumerates entire items of the learning courses.In order to specify the minimum support of the different level of the curriculum scheme, the patterns which are frequently occurred and relatively hard implications are found out. The support value should be given as an absolute significance instead of decimals or percentage. The items that are lower than minimum support value are filtered out which is purely depends upon each level of support threshold. Table 1 refers the user access database which includes the itemset of the curriculum scheme.

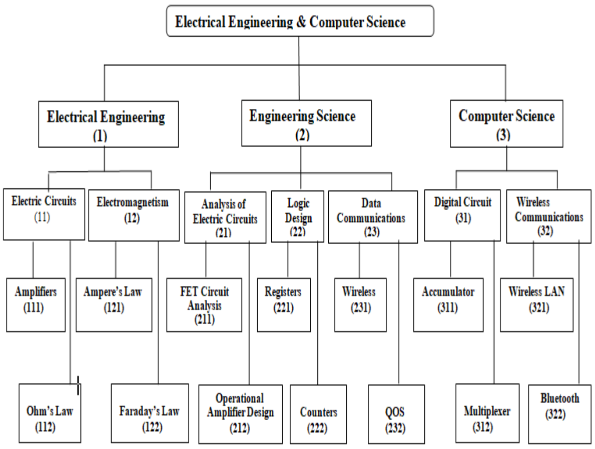


Fig. 3. Concept hierarchy of curriculum scheme

B. FP tree Construction

This phase employs the structure of FP tree helps in compressing the vast database into a extremely reduced; a very small data formation, which obviously avoids expensive and repetitive scanning of database. In accordance with a little sorted order of frequent items, when two or more user access data share a common prefix, the sharing information will be merged via prefix structure method; hence it is easy to build the tree.

An FP tree is represented as a structure of tree that contains a root node referred as NULL, followed by collection of items stated as children of the root called prefix sub-trees; and a frequent item header table. The prefix sub-trees of each node further holds three splitting fileds: the name of the item, count, link of the next node. Count field registers how many transactions has been done and Next node link refers the next node of the FP tree to be executed.

The entry takes place in the frequent item header table mainly has two parts like item-name and head- link where item-name specifies the name of the item and head-link always contains the first node of the FP tree [9] [10].

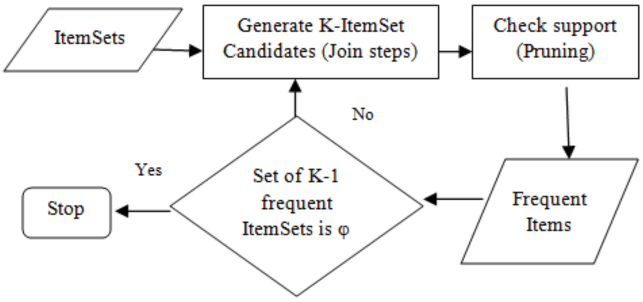

ItemSets
Generate K-ItemSet Candidates (Join steps)
Check support (Pruning)
No
Yes
Stop
Set of K-1 frequent ItemSets is φ
Frequent Items

Fig. 4. Flow chart of FP tree construction algorithm

The flow of constructing an FP tree is described in Fig. 4 where the itemsets are used to compress from large database to small house of data [11]. Here, the construction task is initiated from filtering of data; then the filtered data is passed as parameters for tree construction. The scanning is started once to the access the database followed by collecting the set of frequent itemsets (K) and checks the supports of each level and eliminates the unwanted items by means of pruning. The frequent items are sorted in descending order in order to create the root node of the FP tree (T) indicated as φ.

The frequent items are initially selected, sorted form the enriched database and listed in the order of l. The sorted item list from database is said to be [E/e], where E specifies the first element and e stand for the remaining elements in the list. Using add_tree (P, N) method recursively to insert the corresponding nodes to the tree where P represents the name of the item and N acts as counter. The Next node link helps this algorithm to add the further nodes of the FP tree.

C. Mining Frequent Patterns

This phase completely belongs to mining process which means that the frequent patterns are efficiently mined through FP growth algorithm with the help of FP tree. After mining of the frequent patterns, the generalized association rules are required to generate [12]. Here, the process of mining is started at the base of the header table. Each of the node of the tree is processing out using minimum support count to produce the conditional frequent patterns.

FP growth algorithm is used in this phase for finding the frequent patterns by forming rules using FP tree. Here, the formatted FP tree is passed as input for generating association rules. A method called FPGrowth(tree, set) produced the set of patterns by executing each transactions (T) in FP tree. Before entering to the tree, the root node of the FP tree and the list should be cleared. Next, the conditional patterns are generated from the learning items (set). The algorithm for FP growth is as follows:

```
Procedure: FPGrowth(tree, set)
define and clear F-list :F[]
for each Transaction ($T_i$) in tree do
    for each Item $I_j$ in $T_i$ do
        f[$i_j$]++;
    end
end
sort f[]
define and clear the root of FP-Tree: r
for each Transaction ($T_i$) in tree do
    make $T_i$ ordered according to F:
    call ConstructTree ($T_i$ , r)
end
for each Item $I_j$ in pattern p do
    call Growth (r, $T_{i\ ,}$set)
end
```

D. Pruning

Pruning is the main phase which eliminates the repeated nodes of the FP tree; therefore it improves the efficiency of the FP growth algorithm. By merging the conditional frequent patterns of the FP tree, the frequent accessing rules are formulated. Afterwards, the pruning removed the duplicated items from the frequent access rules.

For every frequent access rules, the related encoded strings are compared excluding the last one. The pruning has to be done once the compared results are identical; else the rules should be retained [13]. As of pruned frequent access rules, the generalized association rules are created for the course learning systems [14].

Furthermore, the remaining gaps of each association rule should be laid down with the interconnection relation of '∩' which ties the related items together. The Association Rule Generation Algorithm is also constructed in this phase from the conditional frequent pattern which is created by FP growth Algorithm which is clearly seen in Table 2. The conditional frequent patterns that are generated from FP growth Algorithm are taken as input for this algorithm.

Apriori algorithm is simply finding frequent items set that satisfy both the minimum support and confidence from the database. This method organizes the improvement of user item from results of frequent itemsets with support of pruning technique. The principle of this a priori is that when the itemset is classified as a frequent itemset, which has more support than previously set, then all subsets are also classed as frequent itemset, and vice versa. Once having a frequent itemset through apriori method, the corresponding rules are consequently produced. Accordance with the calculation of rules with minimum confidence, the rules arrived from the frequent itemset; hence the calculation of support of the itemset is not required. The scanning loop of the database increases many times when we run the apriori algorithm.

FP growth is a substitutionally algorithm which is designed to overcome the drawbacks of Apriori algorithm. It helps in determining the majority frequent items of the itemsets from the course learning database. It follows a special procedure than the method implemented by Apriori. It is an extra ordinary method of finding patterns where the mining is done without any candidate generation. It is capable of building a deep FP tree in order to compact the the original database. Since each transaction in the dataset holds many items, FP growth obviously supports the system of discovering patterns between items in the transactional database. The fundamental thing of association analysis is categorized by two rule evaluation metrics:

Support: It denotes highly occurrence patterns those looking for the items frequently appeared in the dataset within transactions. It is an indication for the items that satisfies the minimum support value in the transactional database. The value of the item support is calucated by

$$Support\ (x \rightarrow y) = \frac{\mathrm{p}\ (\mathrm{x} \cup \mathrm{y})}{\mathrm{N}} \tag{1}$$

Confidence: Once the highly occurence patterns are discovered, then generalized association rules which satisifies the minimum confidence are created. It denotes how frequently the rules have been established to be true. The confidence is defined by the following formula.

$$Confidence\ (x \rightarrow y) = \frac{p\ (x \cup y)}{p\ (x)} \tag{2}$$

where *x*, *y* are the itemsets, N refers the total number of transactions. $p(x \cup y)$ denotes the items appear both in x and y. $p(x)$ specifies the items only in *x* not in *y*.

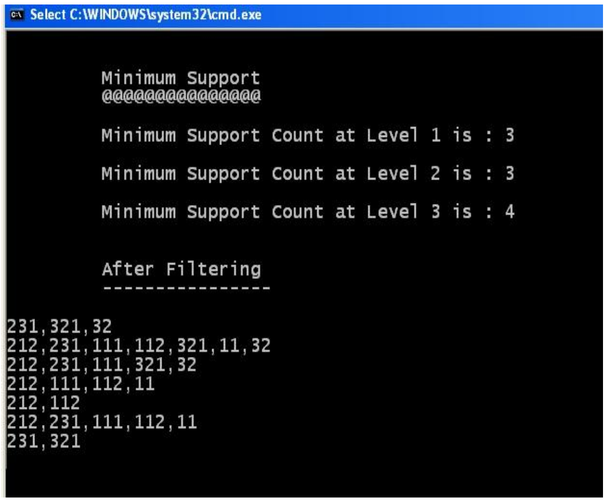
Table- II: Association rule generation

Conditional Frequent Pattern	Frequent access rules	Pruned frequent access rules	Association rules
{(231:3,321:3)}|32	231,321 231,32 321,32 231,321,32	231,321 231,32 231,321,32	231→321 231→32 231→321∩32 231∩321→32
{(212:3,111:3,112:3)}|11	212,111 212,112 212,11 111,112 111,11 112,11 212,111,112 212,111,11 212,112,11 111,112,11 212,111,112,11	212,111 212,112 212,11 212,111,112 212,111,11 212,112,11 212,111,112,11	212→111 212→112 212 11 212→111∩112 212∩111→112 212→ 111 11 212∩111 → 11 212→112∩11 212∩112 →11 212 → 111∩112∩11 212 ∩111 → 112∩11 212∩111∩112 → 11
{(231:4)}|321	231,321	ϕ	ϕ
{(212:4,111:3)}|112	212,111,112	ϕ	ϕ
{(212:3,231:3)}|111	212,231 212,111 231,111 212,231,111	212,231 231,111 212,231,111	212→231 231 →111 212→231∩111 212∩231 → 111
{(212:3)}|231	212,231	ϕ	ϕ

There is an implication relation to be inserted to make it as a meaningful one. So, the gaps between patterns are laid down by an interconnection '∩' and the association rules are formed from pruned frequent access rules.

III. RESULT AND DISCUSSION

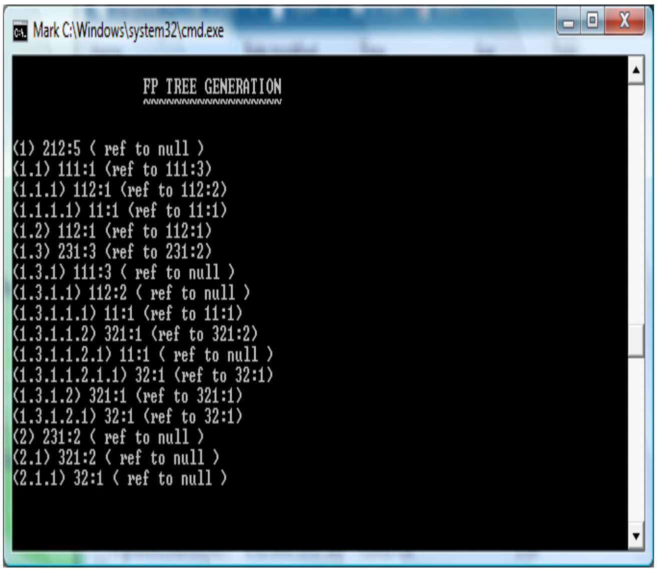
Python is a java based programming language which is especially designed for data analysis and visualization. The implementation of work in this paper is carried out through this language. Before filtering of data, user access database is transformed to enriched database. According to the minimum support of concept hierarchical level, the enriched data is to be filtered. The filtered data is displayed in Fig. 5 with its individual support count.



```
Select C:\WINDOWS\system32\cmd.exe

        Minimum Support
        @@@@@@@@@@@@@@@

        Minimum Support Count at Level 1 is : 3

        Minimum Support Count at Level 2 is : 3

        Minimum Support Count at Level 3 is : 4


        After Filtering
        ---------------

231,321,32
212,231,111,112,321,11,32
212,231,111,321,32
212,111,112,11
212,112
212,231,111,112,11
231,321
```

Fig. 5. Filtering of data

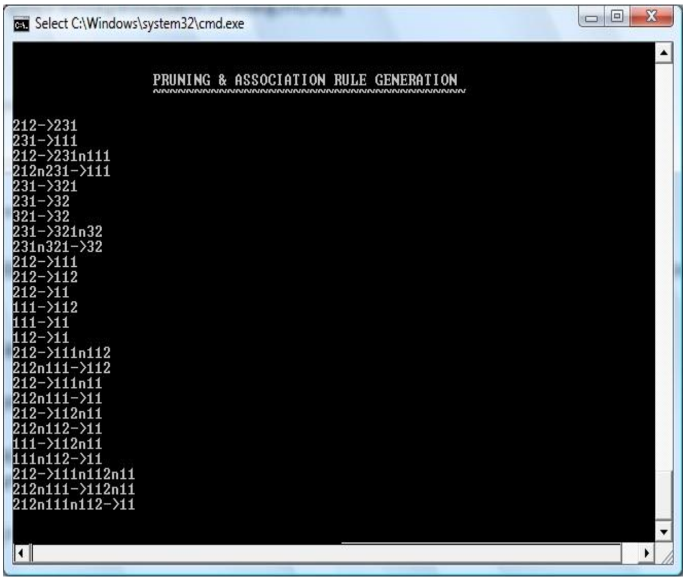
Next process is to construct the FP tree as shown in Fig. 6 by means of filtered data. Through FP tree, generating association rules becomes very easier. Pruning step has also been done here to remove the unwanted branches or nodes while building tree.



```
Mark C:\Windows\system32\cmd.exe

            FP TREE GENERATION
            ^^^^^^^^^^^^^^^^^^

(1) 212:5 ( ref to null )
(1.1) 111:1 (ref to 111:3)
(1.1.1) 112:1 (ref to 112:2)
(1.1.1.1) 11:1 (ref to 11:1)
(1.2) 112:1 (ref to 112:1)
(1.3) 231:3 (ref to 231:2)
(1.3.1) 111:3 ( ref to null )
(1.3.1.1) 112:2 ( ref to null )
(1.3.1.1.1) 11:1 (ref to 11:1)
(1.3.1.1.2) 321:1 (ref to 321:2)
(1.3.1.1.2.1) 11:1 ( ref to null )
(1.3.1.1.2.1.1) 32:1 (ref to 32:1)
(1.3.1.2) 321:1 (ref to 321:1)
(1.3.1.2.1) 32:1 (ref to 32:1)
(2) 231:2 ( ref to null )
(2.1) 321:2 ( ref to null )
(2.1.1) 32:1 ( ref to null )
```

Fig. 6. Construction of FP tree

The frequent patterns from learning dataset are mined efficiently through generalized association rules that are displayed in Fig. 7.

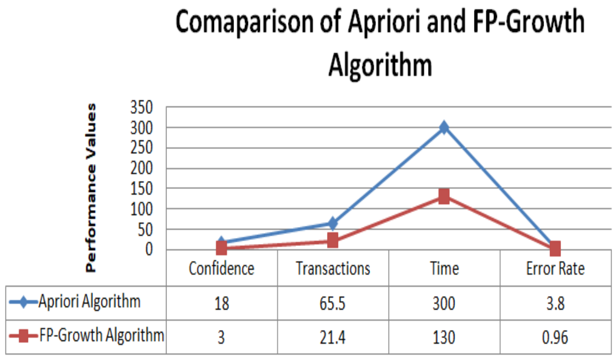


```
Select C:\Windows\system32\cmd.exe

            PRUNING & ASSOCIATION RULE GENERATION
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

212->231
231->111
212->231n111
212n231->111
231->321
231->32
321->32
231->321n32
231n321->32
212->111
212->112
212->11
111->112
111->11
112->11
212->111n112
212n111->112
212->111n11
212n111->11
212->112n11
212n112->11
111->112n11
111n112->11
212->111n112n11
212n111->112n11
212n111n112->11
```

Fig. 7. Pruning and generating association rule

The apriori algorithm scans the dataset repeatedly whereas the FP growth avoids the costly candidate set procedure and generates the highly condensed database. With the help of frequent pattern growth methods, many interesting patterns from learning dataset are mined efficiently as compared with conventional apriori algorithm.

Fig. 8 shows that the level of confidence and transactions are seems to be flexible in FP-Growth, hence the time and error rate are also reduced when judged against apriori algorithm.



	Confidence	Transactions	Time	Error Rate
Apriori Algorithm	18	65.5	300	3.8
FP-Growth Algorithm	3	21.4	130	0.96

Fig. 8. Performance measures of Apriori and FP growth algorithm

IV. CONCLUSION

This work proposed an improved frequent mining algorithm in seeking the frequent patterns in course learning curriculum scheme. According to the performance result, FP growth algorithm is highly efficient in mining the frequent patterns; less error rate and less time to execute the dataset than the apriori algorithm. Even though, FP growth method discovered the competent mining of frequent patterns using generalized association rules, FP tree probably may not robust within memory in some cases, Hence, it is expensive to build and explosive quantity of lacks in good candidate generation method. Future research can combine FP tree with candidate generation method to resolve the existing demerits.

ACKNOWLEDGMENT

The first author would like to thank the management of Kalasalingam Academy of Research and Education for providing fellowship to carry out the research work.

REFERENCES

1. Nguyen, L.T., Vu, V.V., Lam, M.T., Duong, T.T., Manh, L.T., Nguyen, T.T., Vo, B. and Fujita, H., An efficient method for mining high utility closed itemsets. Information Sciences, Vol. 495, 2019, pp.78-99.
2. Pan, Z., Liu, P. and Yi, J., An Improved FP tree Algorithm for Mining Maximal Frequent Patterns. In 2018 10th International Conference on Measuring Technology and Mechatronics Automation , 2018, pp. 309-312
3. Brilliant, M. and Handoko, D., April. Implementation of Data Mining Using Association Rules for Transactional Data Analysis. In Prosiding International conference on Information Technology and Business, 2018, pp. 177-180.
4. Siahaan, A.P.U., Ikhwan, A. and Aryza, S., A Novelty of Data Mining for Promoting Education based on FP growth Algorithm. International Journal of Civil Engineering and Technology, Vol. 9(7), 2018, pp.1660-1669.
5. Borah, A. and Nath, B., FP tree and its variants: towards solving the pattern mining challenges. In Proceedings of First International Conference on Smart System, Innovations and Computing, Springer, Singapore, 2018, pp. 535-543.
6. Heaton, J., Comparing dataset characteristics that favor the Apriori, Eclat or FP growth frequent itemset mining algorithms. In SoutheastCon, 2016, pp. 1-7.
7. Zeng, Y., Yin, S., Liu, J. and Zhang, M., Research of improved FP growth algorithm in association rules mining. Scientific Programming, 2015, pp. 1-6.
8. Bhise, S. and Kale, S., Efficient algorithms to find frequent itemset using data mining. International Research Journal of Engineering and Technology, Vol. 4(6), 2017, pp. 2645-2648.
9. Wu, H., Lu, Z., Pan, L., Xu, R. and Jiang, W., An improved apriori-based algorithm for association rules mining. In 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2, 2009, pp. 51-55.
10. Narvekar, M. and Syed, S.F., An optimized algorithm for association rule mining using FP tree. Procedia Computer Science, Vol. 45, 2015, pp.101-110.
11. Tate, A. and Bewoor, L., January. Survey on frequent pattern mining algorithm for Kernel Trace. In 2017 IEEE 7th International Advance Computing Conference, 2017, pp. 793-798.
12. Krupali, R. and Garg, D., Survey on the Techniques of FP growth Tree for Efficient Frequent Item-set Mining. International Journal of Computer Applications, Vol. 160(1), 2017, Vol. 160(1), pp.39-43.
13. Badhon, B., Kabir, M.M.J., Xu, S. and Kabir, M., A survey on association rule mining based on evolutionary algorithms. International Journal of Computers and Applications, 2019, DOI: 10.1080/1206212X.2019.1612993
14. Sakthivelan, R.G., Rjendran, P. and Thangavel, M., A video analysis on user feedback based recommendation using A-FP hybrid algorithm. Multimedia Tools and Applications, 2019, https://doi.org/10.1007/s11042-019-7293-1.

AUTHORS PROFILE

R. Revathy received B. Sc, M. Sc and M. Phil in computer science from Madurai Kamarajar University, Tamil Nadu. She is pursuing Ph.d in the department of computer applications at Kalsaslingam Academy of Research and Education, Krishnankoil, Tamil Nadu since 2017. Her current research interests include data mining and machine learning algorithms.

S. Balamurali is a Professor of Statistics and Director of Computer Applications at the Kalasalingam Academy of Research and Education. He received his undergraduate, postgraduate and doctoral degrees in Statistics from Bharathiar University, India. His research interests include applied statistics, data mining, network security and bioinformatics.