Design and Implementation of Novel Algorithms for Frequent Pattern Trees

R. Siva Rama Prasad, N.S. Kalyan Chakravarthy, and D. Bujji Babu

1 Introduction

During the past decade, the entire business scenario has been tremendously changed, because of rapid change in the business policies and the organization standards. The ultimate change of their objectivity is to retain customers and the business. The innovations and globalization have generated great opportunities in business and choices in the universal marketplace for firms and customers. The organizations began understanding their customers and their interest to fulfill the requirements of them. As a part of understanding the customers entirely, the organizations are collecting various details of the customers for analysis purpose. The collected data is primarily stored in the fashion of a database, and it allocates a unique identification number to the customer. This kind of primary information is useful to the organizations in maintaining the relationship with the customers. In Data Mining Association rule learning is a popular and well researched technique for discovering interesting relations between variables in large databases. Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence [1][2][3]. Association rules can be extracted using two familiarized algorithms named as Apriori algorithm and FP-Growth algorithm[15][16][17]. The FP-Growth algorithm is completely depends on fp-tree[4]. The previous fp-tree node is labeled only with its current support count, which consumes more time while traversing to extract association rule. In this paper we are more concentrated on design and implementation of novel algorithms for frequent pattern trees. By using the proposed algorithms the traversal time is reduced. In this paper we present six efficient algorithms.

A. ***Frequent Item or Frequent Item set***
An item is said to be a frequent, if it is purchased by minimum number of customers. In other words in any transactional database DB contains any item repeated occurred then the particular item is frequent item. If set of items are repeatedly occurred then that set is called a frequent item set. Generally, these frequent items can be determined using the statistical approach mode. If the data set has only one mode then that is unimodality, if it has two modes the bimodality, if it has three modes then it is called trimodality.

B. ***Association Rule :***
Let D be the database of transactions and ITEMSET = {I1,I2,I3,.. …, In} be the set of items. TR is a transaction includes one or more items in ITEMSET (i.e., TR $\subseteq$ ITEMSET). An association rule has the form P $\Rightarrow$ Q, where P and Q are non-empty sets of items that is P $\subseteq$ ITEMSET, Q $\subseteq$ ITEMSET such that P $\cap$ Q = Ø.

2 Literature Review

A. ***Association Rule Mining(ARM) :***
Agarwal et al.(1993)[1] reported frequent item set mining and association rule mining first time. One of the first algorithms proposed for association rule mining was the AIS algorithm. The Apriori algorithm employs the downward closure property that is if an item set is not frequent any superset of it cannot be a frequent. FP-Growth was developed by Grahne & Zhu and it uses an extra array-based structure to decrease the number of traversals of the tree that are required during the analysis. This saves time during general traversal of the tree and also enables direct initialization of the next level of the

R.S.R. Prasad (✉)
Dept. of I.B Studies, Acharya Nagarjuna University, Guntur, A.P, India
e-mail: raminenisivaram@yahoo.co.in

N.S.K. Chakravarthy
QIS College of Engineering & Technology, Ongole, A.P, India
e-mail: suryaetg@yahoo.com

D.B. Babu
Dept. of Computer Science and Engineering, Prakasam Engineering College, Kandukur, A.P, India
e-mail: bujjibict@gmail.com

FP-Tree(s) [A. Ceglar & J. F. Roddick, 2006; G. Grahne & J. Zhu, 2003]. Another FP-Growth based approach is TD-FP-Growth proposed by Wang et. al., and is a top-down variation to the base FP-Growth approach. This approach is said to alleviate the need or demand to generate/build conditional pattern bases and physical projections of the trie [A. Ceglar & J. F. Roddick, 2006; K. Wang et. al., 2002[5][6][7][8][9]].

B. ***Trees :***

James Joseph Sylvester introduced the term graph in 1878 [10]. Arthur Cayley conducted a study on a particular type of graphs which are called Trees. The study causes several implications in theoretical chemistry. These results were published by George Polya in 1935. Denes Konig wrote the first text book on the graph theory in 1936. In 1962, AVL Trees was invented by G.M.Adelson, Velskii and E.M.Landis[11]. Heinrich Heesch, in 1969 published a computerised problem solving method. Rudolf Bayer and Edward.M first described the B Trees in 1972.

3 Research Objective

The main objective of the research is to reduce the tree traversal time of a frequent pattern tree. Design and implementation of novel algorithms for new frequent pattern tree with new naming approach. To prove how the novel algorithms are effectively working with an example.

4 Proposed Algorithms

A. Algorithm for fp-tree construction:

```
Algorithm   : Novel Fp_tree construction
Input       : Trans_list database DB
Output      : Frequent pattern tree
  1. Read Transaction_list into Trans_list from database
     DB.
  2. Let Freq_items be the set of frequent items
     based on threshold value in Trans_list.
  3. Sort Freq_items on descending order of
     support_count and let the result be F_Sort_List.
  4. Create a root node of an FP_Tree T for
     Trans_list and label it as 'NULL'.
     For each transaction Trans in Trans_list do the
following
     a) Select the frequent items from Trans and the result
        be F_items.
     b) Sort F_items according to the order of F_Sort_List
        and the result be h|R, where h is the first (head)
        element and R is the remaining elements of the list.
  Call insert_tree(h|R,T).
```

B. Proposed Algorithm for insert_tree with Two Level Node Labeling

```
  Algorithm  : insert_tree with Two Level Node Labeling
  Input      : each transaction and Tree
  Output     : Frequent pattern tree with two level node
labels
  1. If T has a child N such that N.item-name = h.item-
     name, then increment N 's count by 1;
  2. else
     {
         create a new node N
         N's count initialized to 1,
     }
  3. N's pred data = T's present data
  4. N's parent link linked to T
  5. N's node-link linked to the nodes with the same item-
     name via the node-link structure.
  6. If P is nonempty, then call insert tree(R, N ) recursively.
```

C. Novel Algorithm to extract n frequent items from FP_growth tree

```
  Algorithm  : NovelFP-growth
  Input      : Tree, Key element a and n.
  Output     : n frequent patterns associated with the
given key element
  Procedure NovelFP-growth(Tree, a,n)
  {
      //a is the key item
    //n indicates the no of items associated with a particular
item.
  Step 1: call Traversal_SPP_and_MPP(Tree, a,n)
  Step 2: return(freq pattern set(SPP) ∪ freq pattern
  set(MPPset) ∪ (freq pattern set(SPP) × freq pattern set
(MPPset)))
  }
```

D. Proposed Algorithm NovelFP-growthSpp

```
  Note  : X represents a pattern in single predecessor path
and Y represents a pattern in Multi predecessor path.
  Algorithm  :  NovelFPgrowth_to_traverse_SinglePre-
decessor Path
  Input      : Tree, Key element a and n.
  Output     : Traverses in single predecessor path and
returns the patterns associated with the key
  NovelFP-growthSpp(Tree, a,n)
  {
  Step 1: let SPP be the Single Predecessors path of n-1
nodes part of Tree;
  Step 2: for each combination (denoted as X) of the nodes
in the path SPP do
  Step 3: generate pattern X ∪ a with support = minimum
support of nodes in X;
  Step 4: let freq pattern set(SPPSet) be the set of patterns
so generated;
  }
```

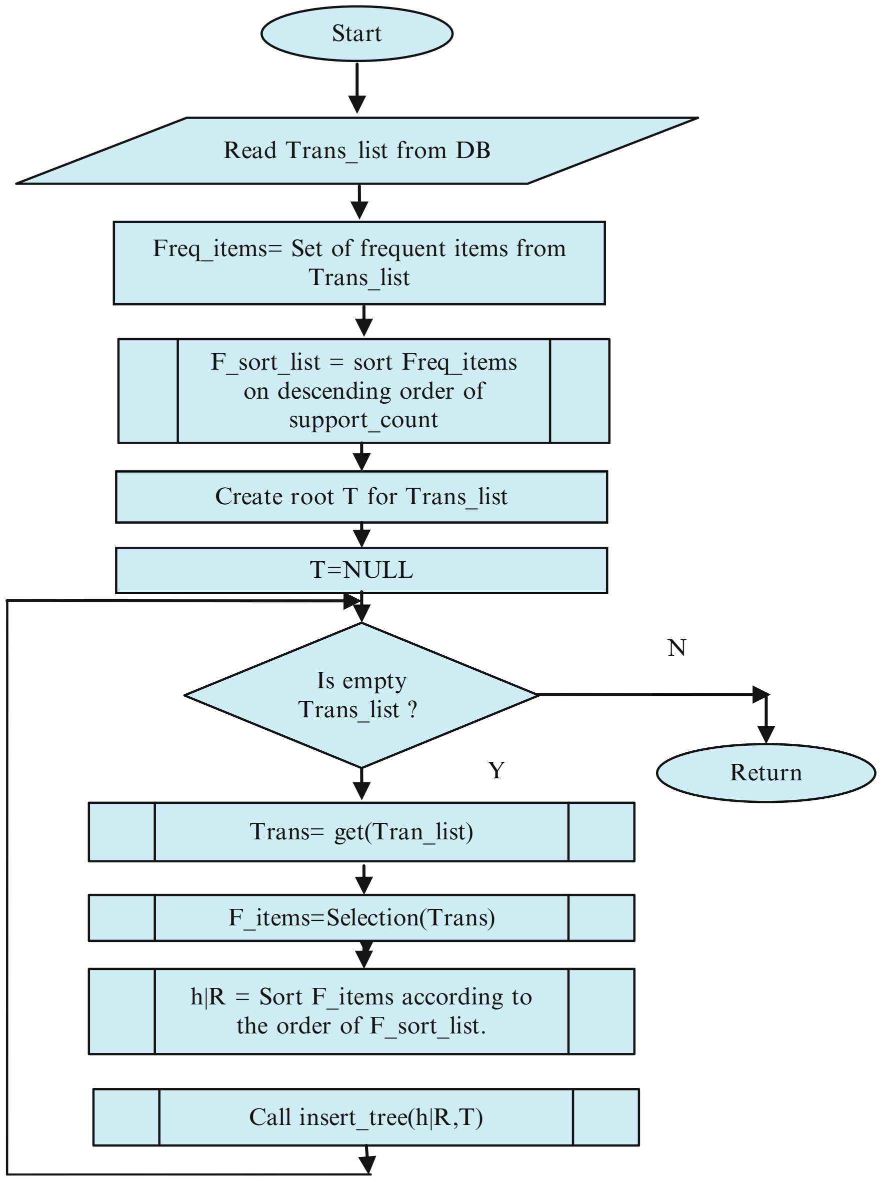

Start
Read Trans_list from DB
Freq_items= Set of frequent items from Trans_list
F_sort_list = sort Freq_items on descending order of support_count
Create root T for Trans_list
T=NULL
Is empty Trans_list ?
N
Return
Y
Trans= get(Tran_list)
F_items=Selection(Trans)
h|R = Sort F_items according to the order of F_sort_list.
Call insert_tree(h|R,T)

Figure: 1 Flow chart for Novel Fp_tree construction Algorithm

E. Proposed Algorithm NovelFP-growthMpp

Algorithm : NovelFP-growth_to_traverse_MultiPredecessor Path

Input : Tree, Key element a and n.

Output : Traverses in multi predecessor path and returns the patterns associated with the key

```
NovelFP-growthMpp(Tree, a,n)
{
1. let MPP be the Multi Predecessors Path part of Tree with i=1;
2. for each item $a_i$ in MPP with i <= n-1 do
   {
3. generate pattern Y = ai ∪ a with support = ai.support;
4. i=i+1
   }
5. construct Y's conditional pattern-base and then Y's conditional FP-tree  Tree Y;
6. if Tree Y $\neq$ Ø then
7. call NovelFP-growth( Y, a, n);
8. let freq pattern set(MPPSet) be the set of patterns so generated;
}
```

F. Proposed Algorithm Traversal_SPP_and_MPP

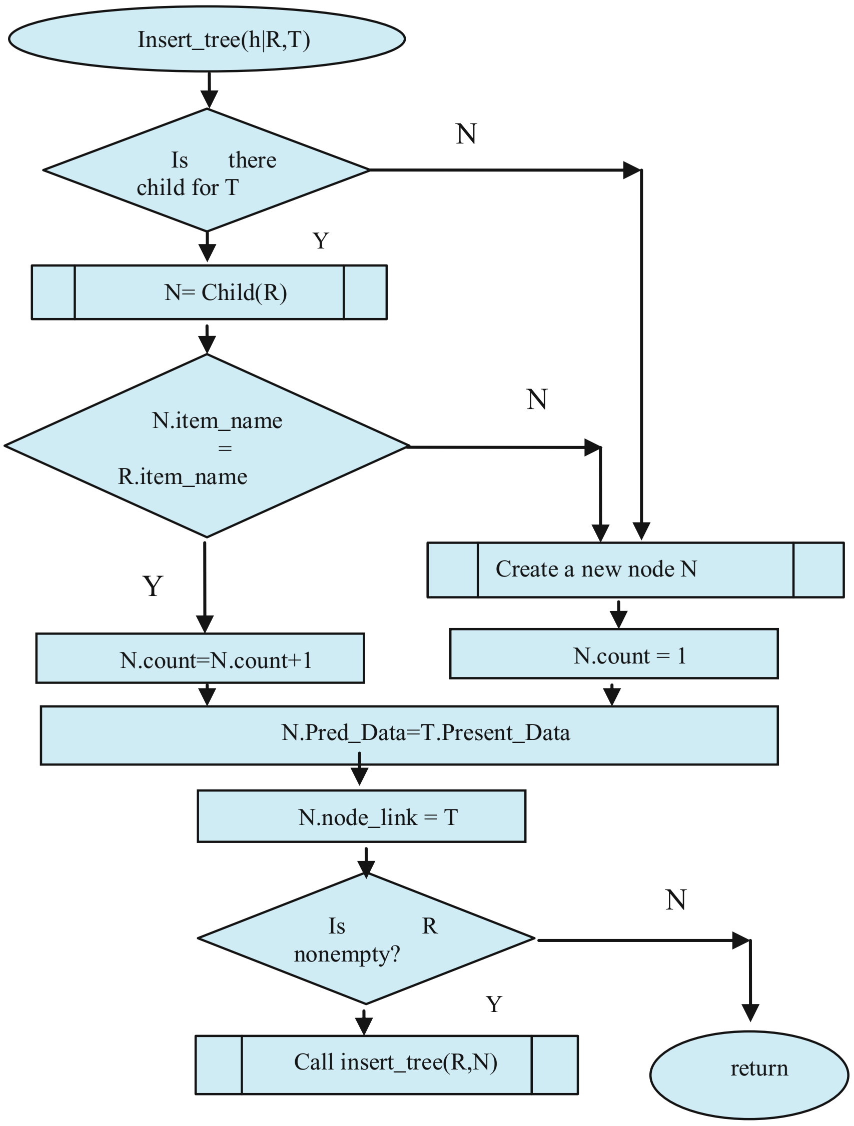
Algorithm : Traversal_SPP_and_MPP

Input : Tree, Key element a and n.

Output : Traverses either in single predecessor path or in multi predecessor path and returns the patterns associated with the key

//Freq pattern set SPPSet and MPPSet are global variables.

Figure: 2 Flow chart for insert_tree with Two Level Node Labeling Algorithm



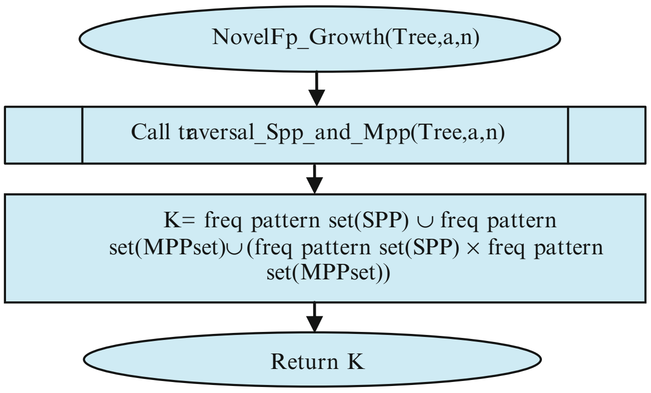

NovelFp_Growth(Tree,a,n)
Call traversal_Spp_and_Mpp(Tree,a,n)
K= freq pattern set(SPP) ∪ freq pattern set(MPPset)∪(freq pattern set(SPP) × freq pattern set(MPPset))
Return K

Figure: 3 Flow chart for NovelFP-growth Algorithm

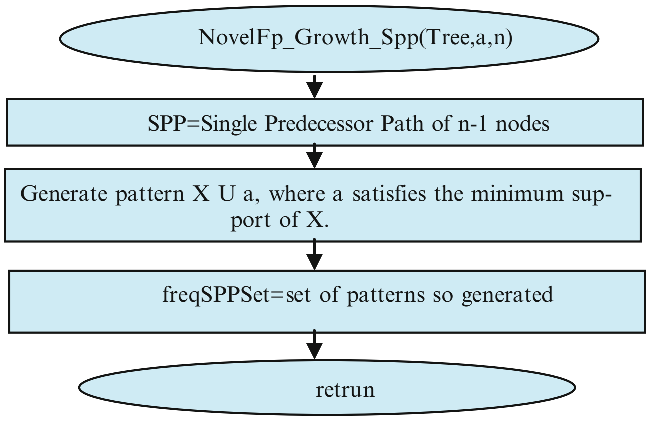
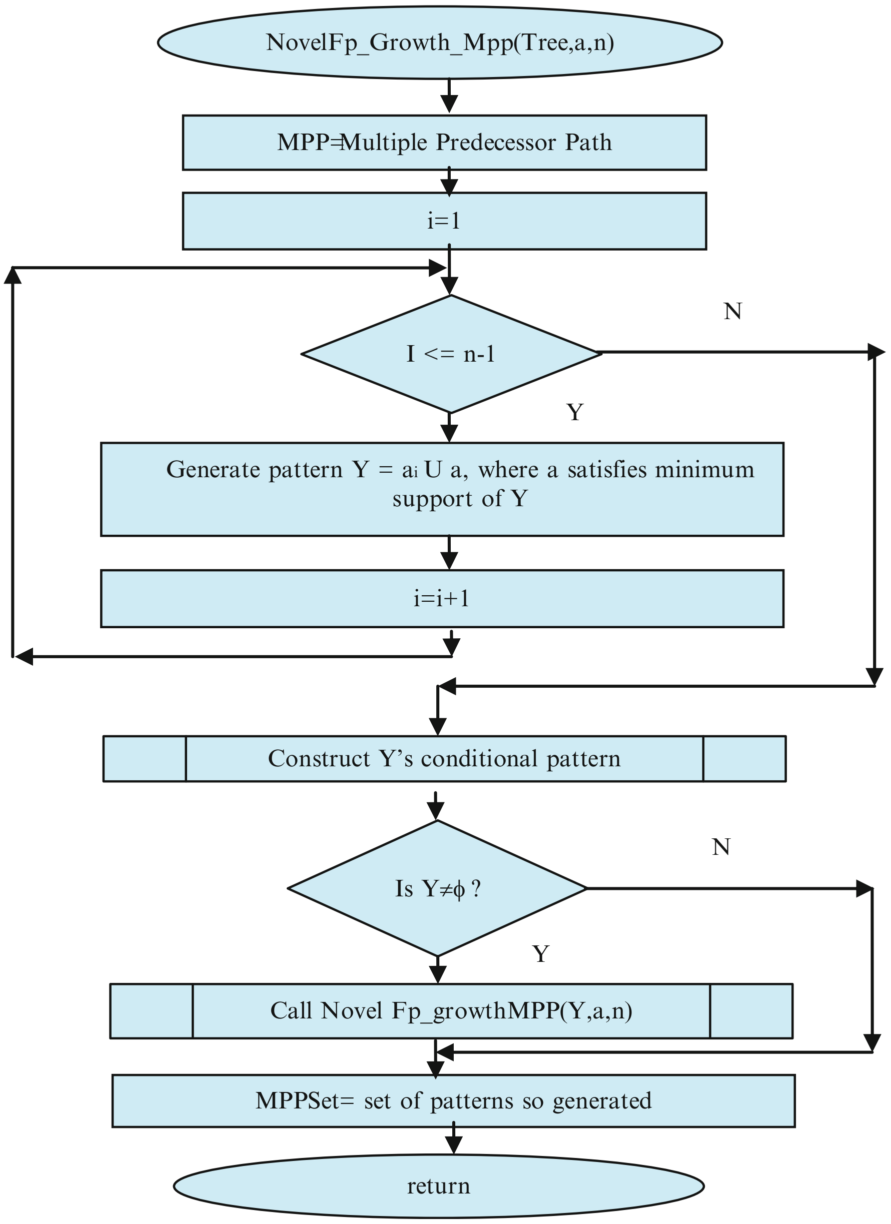

NovelFp_Growth_Spp(Tree,a,n)
SPP=Single Predecessor Path of n-1 nodes
Generate pattern X U a, where a satisfies the minimum support of X.
freqSPPSet=set of patterns so generated
retrun

Figure: 4 Flow chart for NovelFP- growth_to_traverse _SinglePredecessorPath

Figure: 5 Flow chart for NovelFP-growth_to_traverse_ MultiPredecessorPath Algorithm


NovelFp_Growth_Mpp(Tree,a,n)
MPP=Multiple Predecessor Path
i=1
I <= n-1
N
Y
Generate pattern Y = ai U a, where a satisfies minimum support of Y
i=i+1
Construct Y's conditional pattern
Is Y≠ϕ ?
N
Y
Call Novel Fp_growthMPP(Y,a,n)
MPPSet= set of patterns so generated
return

```
    Procedure Traversal_SPP_and_MPP(Tree, a,n)
   {
  //n indicates the no of items associated with a particular item.
  Step 1: if Tree contains a single Predecessors path then
            Call NovelFP-growthSpp(Tree, a,n)
        else
            Call NovelFP-growthMpp(Tree, a,n)
    Step 2: return(freq pattern set(SPPSet) ∪ freq pattern set(MPPset)
        ∪ (freq pattern set(SPP) × freq pattern set (MPPset)))
   }
```

5 Performance and results

Theorem : The time complexity of The Two-level node labeling algorithm(s) is O(n) with reduction of one unit of time in worst case also.

Proof : The proposed Two Level Node Labeling algorithm constructs a frequent pattern tree which is similar to binary tree. Hence, the time complexity of traversal of frequent pattern tree, which is formed with the proposed algorithm, is similar to the traversal complexity of a binary

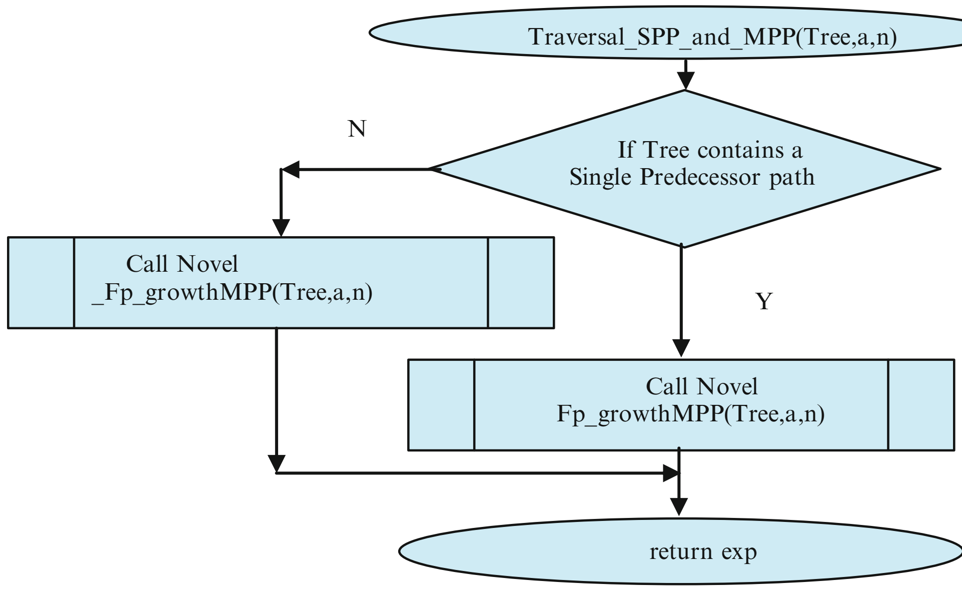
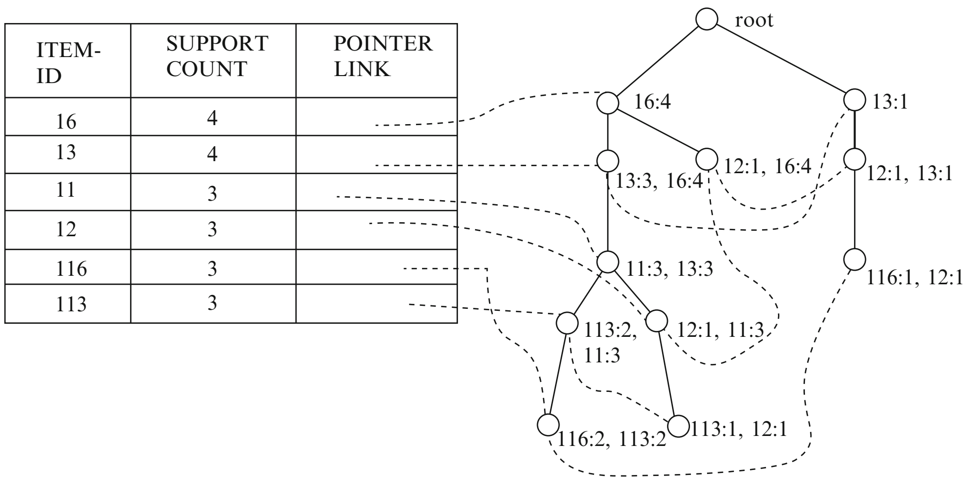

Traversal_SPP_and_MPP(Tree,a,n)
If Tree contains a Single Predecessor path
N
Y
Call Novel _Fp_growthMPP(Tree,a,n)
Call Novel Fp_growthMPP(Tree,a,n)
return exp

Figure: 6 **Flow chart for Traversal_SPP_and_MPP Algorithm**



ITEM-ID	SUPPORT COUNT	POINTER LINK
16	4	
13	4	
11	3	
12	3	
116	3	
113	3	

Figure: 7 **Frequent item Transactional Data Base**

Table:1 **Transactional Data Base**

TID	ITEMS PURCHASED
T100	I6, I1, I3, I4, I7, I9, I13, I16
T200	I1, I2, I3, I6, I12, I13, I15
T300	I2, I6, I8, I10, I15
T400	I2, I3, I11,I17, I16
T500	I1, I6, I3, I5, I9, I16, I13, I14

Table:2 **Frequent item Transactional Data Base**

TID	ITEMS PURCHASED
T100	I6, I3, I1, I13, I16
T200	I6, I3, I1, I2, I13
T300	I6, I2
T400	I3, I2, I16
T500	I6, I3, I1, I13, I16

tree. Since the time complexity of binary tree traversal algorithm is O(n) [13][14], the time complexity of frequent pattern tree is O(n). But the proposed algorithm reduces one level in the tree during traversal.

With taking the training data set as in the table:1 [12]. We get the frequent transactional data base as shown in the table:2 with considering the minimum support value as 3. The final frequent pattern tree will appears as shown in figure:7 by using the proposed algorithms.

The traversal time for the predecessor nodes in frequent pattern trees[18][19][20][21] using the traditional frequent pattern tree approach is only O(n) for the nth level node. The proposed algorithm also constructs a binary tree. Hence the time complexity of the newly constructed tree is also O(n). But, one unit of time is less than the time of traditional approach in worst case. The proposed algorithm takes a fewer number of node comparisons to determine the predecessor nodes and its path than that of traditional approach in best and average cases.

6 Conclusion

In this paper, we presented the design and implementation issues of six novel algorithms. One algorithm constructs a frequent pattern tree. Four algorithms are used to label each node and another algorithm extracts the frequent patterns from the frequent pattern tree. In the Example figures the solid line between the nodes represents the relation between the nodes and the dashed line indicates the pointer link between the same nodes, useful to maintain the cumulative node count in the data structure. We also presented the flow charts of each algorithm.This novel approach reduce one level tree traversal of the tree in the worst case also.

Acknowledgements We are so grateful to Sri. Dr.Kancharla Ramaiah garu,Secretary and correspondent of Prakasam Engineering College, kandukur, for extending his marvelous encouragement and support to do the research with providing the research environment. Last but not least, we are very much thankful to all the authors and co-authors of the reference papers for providing us knowledge about clouds, cloud environment and the data mining techniques particularly about association rule learning process and algorithms.

References

1. Agarwal, R., and Srikanth,R., "Fast Algorithms for mining association rules," In Proc. Of the International Conference on VLDB-94, Sept.1994,pp.487-499.
2. T.Mitchell." Machine learning," Mc Graw Hill, Boston,M.A,1997.
3. J.Han and m.Kamber. "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers, San Francisco, 2001.
4. Pang-ning-Tan,Vipin Kumar,Michael Steinbach." Introduction to Data Mining" Pearson 2007. ISBN 978-81-317-1472-0.
5. Bodon. F, "A Survey on Frequent Itemset Mining", Technical report, Budapest Univ. Of Technology and Economics, 2006.
6. Cheung D, V.T Ng, A. Fu, and Y.Fu. "Efficient mining of association rules in distributed databases". *IEEE Trans. Knowledge and Data Engineering*, pp 1-23, 1996
7. Rupali Haldulakar and Prof. Jitendra Agrawal, "Optimization of Association Rule Mining through Genetic Algorithm", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 3 Mar 2011, pp. 1252-1259.
8. Manish Saggar, Ashish Kumar Agarwal and Abhimunya Lad, "Optimization of Association Rule Mining using Improved Genetic Algorithms"IEEE 2004.
9. Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K., "Optimized association rule mining using genetic algorithm", Advances in Information Mining, ISSN: 0975–3265, Volume 1, Issue 2, 2009, pp-01-04.
10. J. J. Sylvester (1878) "On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics, — with three appendices," *American Journal of Mathematics, Pure and Applied*, (1) : 64-90. The term "graph" first appears in this paper on page 65.
11. Adelson-Velskii and E. M. Landis, 1962. "An algorithm for the organization of information". *Proceedings of the USSR Academy of Sciences* 146: 263–266. (Russian) English translation by Myron J. Ricci in *Soviet Math. Doklady*, 3:1259–1263, 1962.
12. D.Bujji Babu,Dr.R.Sivarama Prasad and Y.Umamaheswararao "Efficient Frequent Pattern Tree Construction",*International Journal of Advanced Computer Research, Volume -4 Number-1issue-14 March-2014*.
13. http://www.geeksforgeeks.org/618
14. Massachusetts Institute of Technology (MIT), "Master Theorem: Practice Problems and Solutions", http://www.csail.mit.edu/~thies/6.046-web/master.pdf.
15. M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li.New Algorithms for Fast Discovery of Association Rules. *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97)*, 283–296. AAAI Press, Menlo Park, CA, USA 1997.
16. G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *FIMI'03, Workshop onFrequent Itemset Mining Implementations*, November 2003.
17. F. Masseglia, F. Cathala, and P. Poncelet. Psp : Prefix tree for sequential patterns. In Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98) Nantes France LNAI, pages 176–184, 1998.
18. Adelson-Velskii and E. M. Landis, 1962. "An algorithm for the organization of information". *Proceedings of the USSR Academy of Sciences* 146: 263–266. (Russian) English translation by Myron J. Ricci in *Soviet Math. Doklady*, 3:1259–1263, 1962.
19. F. Masseglia, F. Cathala, and P. Poncelet. Psp : Prefix tree for sequential patterns. In Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98) Nantes France LNAI, pages 176–184, 1998.
20. G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *FIMI'03, Workshop onFrequent Itemset Mining Implementations*, November 2003.
21. C. Borgelt. Recursion Pruning for the Apriori Algorithm. *Proc. 2nd IEEE ICDM Workshop onFrequent Item Set Mining Implementations (FIMI 2003, Brighton, United Kingdom)*. CEUR Workshop Proceedings 126, Aachen, Germany 2004.http://www.ceur-ws.org/Vol-126/
22. Leung, C. K. S., Mateo, M. A. F., & Brajczuk, D. A. (2008). A tree-based approach for frequent pattern mining from uncertain data. Lecture Notes in Computer Science, 5012, 653–661.