CIMTEL- Mining Algorithm for Big Data in Telecommunication

G.Nagarajan[#1], R.I.Minu[*2], V.Vedanarayanan[#3] S.D.Sundersingh Jebaseelan[#4], K.Vasanth[#5]

[#]Sathyabama University, Chennai
[#1]nagarajanme@yahoo.co.in
[*]Jerusalem College of Engineering, Chennai
[*2]minu@jerusalemengg.ac.in

Abstract—**T**he field of data mining has flourished into research area of significant technological and social importance due to the advancement in technology. Mining frequent pattern or itemset from real data environment is a fundamental and essential problem in many data mining applications. The Apriori-inspired algorithms show good performance with sparse datasets such as market-basket data, where the frequent patterns are very short. However, in the area with dense datasets such as telecommunication, computational biology and census data, the performance of these algorithms degrades incredibly as there were many, long frequent patterns. The focus of this paper is to design a CPU- efficient algorithm CIMTEL for finding closed frequent calling patterns (long pattern) in a telecommunication database. Due to the evaluation of next generation telecommunication network the amount of communication data's rises in volume, vaiety and velocity (Bid Data). Thus algorithm provides an way for analysis the consumer for service provider. The Performance of this algorithm outperforms the former COLTEL , CHARM and EXPEDITE algorithm by an order of two for a worst case scenario. Also the performance analysis of this algorithm with former algorithms is determined.

Keyword-Big Data, Close Item Set, Patten Mining , Frequent Item Set Mining:

I. INTRODUCTION

The explosive growth in data collection in business and scientific fields has literally forced the need to analyze and mine useful knowledge from it. Data mining refers to the entire process of extracting useful and novel patterns/models from large datasets. It is a process through which interesting and previously unknown patterns and correlations can be extracted automatically from a large database of information. The implicit information within databases, and mainly the interesting association relationships among sets of objects, that lead to association rules, may disclose useful patterns for decision support, financial forecast, marketing policies, medical diagnosis and many other applications.

Association rule mining involves detecting items, which tend to occur together in transactions, and the association rules that relate them. Association rule mining has applications in cross-marketing, attached mailing, add-on sales, store layout and customer segmentation based on buying patterns. Mining frequent itemsets is a fundamental and essential operation in data mining applications including discovery of association rule, strong rules, correlations, sequential rules and episodes. Due to the huge size of data and amount of computation involved in data mining, high performance computing is an essential component for any successful large-scale data mining application. Association Rule mining finds the set of all subsets of items or attributes that frequently occur in many database records or transactions, and additionally extracts rules on how a subset of items influences the presence of another subset.

Consider I = {i1, i2 … im} as a set of items. Let *D*, the task relevant data, is a set of database transactions where each transaction *T* is a set of items such that *T* is a subset of I. Each transaction is associated with an identifier, called TID. Let *A* be a set of items. A transaction *T* is said to contain *A* if and only if *A* is a subset of *T*. An association rule is an implication of the form *A => B*, where *A* and B are subsets of I and A *B* is also a subset of I. The rule A=>B holds in the transaction set D with support s, where *s* is the percentage of transactions in *D* that contain *A* U*B* (i.e., both *A* and *B*). This is the probability, *P* (*A*U*B*). The rule *A =>B* has *confidence c* in the transaction set *D* if *c* is the percentage of transactions in *D* containing *A* that also contain *B*. This is taken to be the conditional probability, *P (B|A)*. That is,

$$Support\ (A \cup B) = P(A \cup B) \quad (1)$$

$$Confidence\ (A \cup B) = P(B \mid A) \quad (2)$$

The definition of a frequent pattern relies on the following considerations. A set of items is referred to as an itemset (pattern). An itemset that contains *k* items is a *k*-itemset. The set {X, Y} is a 2- itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known as the frequency or the support count of an itemset. An itemset satisfies *minimum support* if the occurrence frequency of the itemset is greater than or equal to the *minimal support threshold value* defined by the user. The number of transactions required for the itemset to satisfy *minimum support* is therefore referred to as the *minimum support count*. If an itemset satisfies *minimum support*, then it is a *frequent itemset* (*frequent pattern*). A frequent itemset

is called closed if it does not have any superset with the same support. A frequent itemset is said to be maximal if it has no supersets that are frequent. The collection of maximal frequent itemsets is a subset of the collection of closed frequent itemsets, which is a subset of the collection of all frequent itemsets. Maximal frequent itemsets are necessary for generating association rules.

A. Telecommunication Database

Telecommunication systems was introduced at late 1800's , till now there were so many advancement in this field. Nowadays, most of the telecommunication systems are controlled by computer software's. Basically telephones are connected via Centrex switches, for each call made by an customer is connected to the destination via these switches, which is controlled by computer and they maintain a database which is clearly explained below. So, for the near future database will be the central role in telecommunications networks. The information needed in operations and management of the nets will be collected into a logically uniform database. The world-wide nature of telecommunications prescribes that the only possibility to obtain the logical uniformity is the co-operation of autonomous databases. A telecommunications database system should be able to support short but voluminous simple read transactions, long but voluminous simple updating transactions, and a few very long complex updating transactions in the same real-time database system. Due to the evaluation of next generation telecommunication network the amount of communication data's rises in volume, variety and velocity . These huge advancement transform the telecommunication database to Big data [15]. Thus algorithm provides an way for analysis the consumer for service provider

B. Mining in Telecommunication Database

A telecommunication system however, produces daily a large amount of data which contains hidden valuable information about the calling patterns of consumers. In [TASA] designed a system for discovering and browsing knowledge from telecommunication network alarm databases called TASA (Telecommunication Network Alarm Sequence Analyzer). The system uses a framework for locating frequently occurring episodes from sequential data. In [GSM] the author proposed an algorithm for mining sequential alarm patterns from the alarm data of a GSM system.

A telecommunication network can be viewed as consisting of a number of interconnected components: switches, exchanges, transmission equipment, etc. Each component in its turn contains several sub-components. The number of components depends on the abstraction level used in viewing the system. A network operated by a local telephone company can be considered to contain 10 - 1000 components.

These data include call detail data, which describes the calls that traverse the telecommunication networks, network data, which describes the state of the hardware and software components in the network, and customer data, which describes the telecommunication customers. The amount of data is so great that manual analysis of the data is difficult, if not impossible. The need to handle such large volumes of data led to the development of knowledge-based expert systems. These automated systems performed important functions such as identifying fraudulent phone calls and identifying network faults. The problem with this approach is that it is time consuming to obtain the knowledge from human experts and, in many cases, the experts do not have the requisite knowledge. The advent of data mining technology promised solutions to these problems and for this reason the telecommunications industry was an early adopter of data mining technology.

II. LONG PATTERN MINING

Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes classifiers, clusters and many more of which the mining of association rules is one of the most popular problems.

Most of the proposed pattern-mining algorithms are a variant of Apriori. Apriori employs a bottomup breadth first search that enumerates every single frequent itemset. However, with dense datasets where there are many, long frequent patterns, the performance of these Apriori-inspired algorithm degrades incredibly. This degradation is due to the following reasons: these algorithms perform as many passes over the database as the length of the longest frequent pattern. Secondly, a frequent pattern of length l implies the presence of 2l - 2 additional frequent patterns as well, such algorithms explicitly examine each of which. When l is large, the frequent itemset mining methods become CPU bound rather than I/O bound. In other words, it is practically unfeasible to mine the set of all frequent patterns for other than small l. On the other hand, in many real world problems (e.g., patterns in Telecommunication database).

A-Close [16] was the first algorithm for closed itemset mining based on the Apriori heuristic. A-close is a variation of Apriori, it adopts the Apriori framework, but looks for frequent closed itemsets and prunes the frequent itemsets that are not closed. The major cost of the A-Close is from two aspects: (1) it has to generate a lot of candidates and scan the transaction database again and again to count candidates; and (2) in the last scan to compute closures, there could be a large number of surviving frequent itemsets. For each transaction, the intersection with each surviving frequent itemsets is done. This makes the closure computation quite costly. In [13] the author proposed a algorithm for frequent closed itemsets and frequent generators mining algorithm. In

[15] for frequent itemset mining , they proposed a pattern mining methodology which would detect meaningful and distinct pattern in real data sets.

There are two current solutions to the long pattern mining problem. The first one is to mine only the maximal frequent itemsets , which are typically orders of magnitude fewer than all frequent patterns. While mining maximal sets help understand the long patterns in dense domains, they lead to a loss of information; since subset frequency is not available maximal sets are not suitable for generating rules. The second is to mine only the frequent closed sets . Closed sets are lossless in the sense that they uniquely determine the set of all frequent itemsets and their exact frequency. At the same time closed sets can themselves be orders of magnitude smaller than all frequent sets, especially on dense databases.

A. *Closed Frequent Itemset*

CLOSET [6] for mining closed frequent patterns. This algorithm inherits from FP-growth, the compact FP-Tree data structure and the exploration technique based on recursive conditional projections of the FP-Tree. Frequent single items are detected after a first scan of the dataset, and with another scan, the pruned transactions are inserted in the FP-Tree tored in the main memory. Despite the efficiency of this FP- growth, if the database is huge, the FP-tree will be large and the space requirement for recursion is a challenge [9]. CHARM for finding closed frequent itemsets proposed by Zaki and Hsiao performs a bottom-up depth first browsing of a prefix tree of frequent itemsets built incrementally. As soon as a frequent itemset is generated, its tid-list is compared with those of the other itemsets having the same parent. When two tid-lists are equal, or one includes the other, the associated nodes are merged since the itemsets surely belong to the same equivalence class. Itemset tid-lists are stored in each node by using the diff-set technique [10]. In [12] the author design a PCA based mining algorithm for online streaming mining. EXPEDITE [14] this algorithm are used for effective frequent closed itemset mining which provides a CPU time saving strategies. The time required to mine the intermediate item sets are reduced dramatically in this mining algorithm.

A frequent itemset, X is now said to be a closed itemset if there exists no X' such that X' is a proper superset of X and every transaction containing X also contains X'. A closed itemset is frequent if it passes the given support threshold. Knowledge about closed frequent patterns is interesting and useful when the right algorithm is used. Another approach of mining closed frequent patterns which adopts the methodology of pattern growth methods and avoids the candidate generation and test approach is hereby proposed. A pattern growth uses the Apriori property, however, instead of generating candidates-sets, it recursively partitions the database into sub-databases according to the frequent patterns found and searches for local frequent patterns to assemble local ones.

The scenario below gives a good understanding of closed frequent patterns.

Suppose the frequent patterns generated are:

{bread, butter: 10};

{sugar, butter: 10};

{bread, sugar : 10};

{bread, sugar, butter: 10}.

Closed frequent pattern mining will return one itemset only:

{bread, sugar, butter: 10}.

This itemset, however, represents the complete information about the frequency of its three sub-itemsets.

III. CIMTEL

CIMTEL (Closed Itemset Mining in Telecommunication Database) -An efficient algorithm for enumerating the set of all frequent closed itemsets. Some of the innovative ideas employed for the development of the algorithm:

1. They simultaneously explore both the itemset space and transaction space over a novel IT-tree search space. In contrast, most previous methods exploit only the itemset search space.
2. They use a highly efficient hybrid search method that skips many levels of the IT tree to quickly identify the frequent closed itemsets, instead of having to enumerate many possible subsets.
3. By arranging the items in decreasing order we can determine all the closed itemsets within first two complete iteration .

A. *Algorithm Design*

Given the CIMTEL, an efficient algorithm for mining all the closed frequent itemsets. We will first describe the algorithm in general terms, independent of the implementation details. We then show how the algorithm can be implemented efficiently. CIMTEL simultaneously explores both the itemset space and tidset space using the IT-tree, unlike previous methods which typically exploit only the itemset space. CIMTEL uses a novel search

method, based on the IT-pair properties as explained earlier. Rather than having to fully numerate the whole possible subsets of IT- Tree this algorithm skips many levels in the IT-tree to quickly converge on the itemset closures.

Algorithm: CIMTEL (Closed Itemset Mining in Telecommunication Database)

- **Input** → Telecommunication database (D,min_sup)
- **Output** → Closed Frequent Itemsets. (C)
- **Method:**

  Step 1 : Find all the Frequent Itemsets in [F] and call the Gcfi

  Step 2 : Arrange the itemset by descending order

  Step 3 : Compare and check with the closure property to find all the cfi

  Step 4: Delete the current IT pair subsume and recursively repeat for all itemset the step 2 through step 4

Now following shows the pseudo code of the algorithm and follows its explanation:

CIMTEL(D,min_sup)

1. For all transaction in TDB

   If ($X_i \varepsilon I$) Λ ($\sigma(X_i) >= min\_sup$)

   Insert the item into Frequent_itemset projection [F]

   Else if delete X_i

2. Call CIMTEL-Gcfi ([F],C=0)

3. return C *// all closed sets //*

The algorithm starts by initializing the prefix class [F], of nodes to be examined, to the frequent single items and their tidsets in Line 1. After determining all the frequent itemsets at Line 2 the main computation procedure Gcfi is called which returns the set of closed item set at Line 3.

CIMTEL-Gcfi ([F],C)

1. For each item X_i in [F] compute the weight $w(X_i)$ & sort them in decreasing order

   let $X = X_i$

   for each X_{i+1} x $t(X_{i+1})$ in [F]

   $X = X \cup X_i+1$ and $Y = t(X_i) \cap t(X_{i+1})$

   call CIMTEL-Property ([F],[Fi])

2. Recursively call CIMTEL-Gcfi ([F],C) until the current projected itemset is completely compared

3. Delete the current projected itemset list

4. C = All Closed itemsets

At Line 3 the IT pairs which was subsumed by other pair are all deleted to free the memory

Here at Line 1 each item X_i in [F] the weight $w(X_i)$ is computed ,where the weight of an item X is $w(X) = \sum_{xy \in F2} \sigma(XY)$ i.e., the sum of the support of frequent 2-itemsets that contain the item. Then the two IT-pairs are combined to produce a new pair **X** and **Y**, where $\mathbf{X} = X_i \cup X_{i+1}$ and $\mathbf{Y} = t(X_i) \cap t(X_{i+1})$ and the property procedure is called to check for closed itemset. Then through recursive call the complete set of item are projected at its corresponding closed itemset are determined space. Thus at the end C has all the closed frequent itemsets.

CIMTEL-Property ([F],[F_i])

1. if ($s(X) \geq min\_sup$) then

   if $t(X_i) = t(X_{i+1})$ then

   Remove $X_{i+1}$from [F] *// PROPERTY 1 //*

   Replace all X_i with X

2. else if $t(X_i) \subset t(X_{i+1})$ then

Replace all Xi with X *// PROPERTY 2 //*

3. else if $t(X_i) \supset t(X_{i+1})$ then

   Remove $X_{i+1}$from [F] *// PROPERTY 3 //*

   Add X x Y to [F_i] *// by order //*

4. else if $t(X_i) \neq t(X_{i+1})$ then *// PROPERTY 4 //*
   Add X x Y to $[F_i]$ *// by order //*

Property 1 : If the transactions of two itemsets is same, then it means the closure of both the itemsets are also same so we can remove all X_{i+1} as both yield the same result.

Property 2: If the transaction of X_i is included in X_{i+1} then replace all X_i with X (X_i U X_{i+1})

Property 3: If the transaction of X_{i+1} is included in X_i then remove all X_{i+1} and insert them by the decreasing order

Property 4: If both the transaction is not equal then insert it to array in decreasing order

IV. PERFORMANCE ANALYSIS

Fig.1 shows the performance of the algorithm CIMTEL with other algorithm like APRIORI and CHARM. The graph shows the variation in time with the Min_support value. The values for APRIORI and CHARM are gathered by implement a model Telecommunication database in a software called WEKA a Java oriented software. The analysis shows that the time taken by CIMTEL is less when compare with CLOSET,CHARM and EXPEDITE.

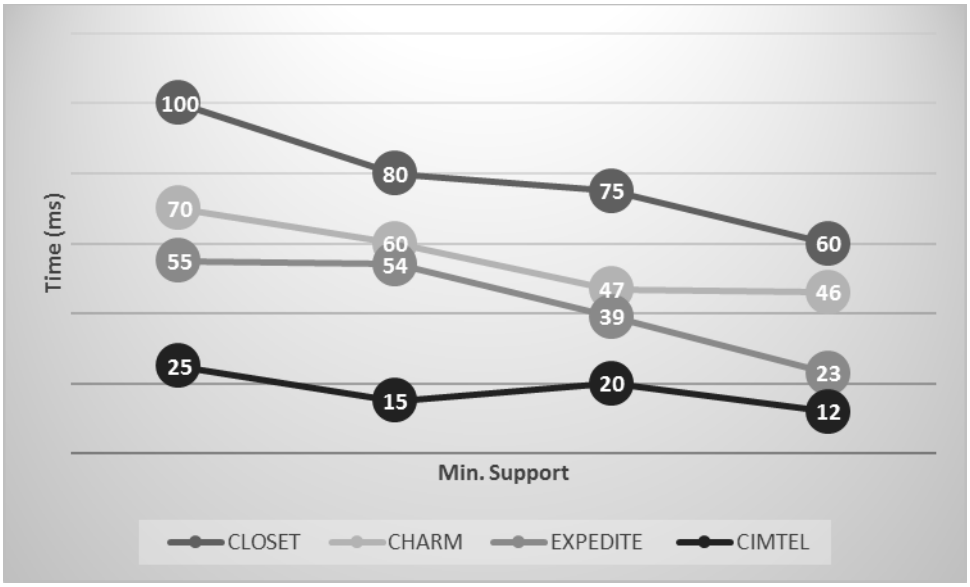

Figure 1 Algorithm Analysis

C. COMPARISON ANALYSIS

The performance of the algorithm CIMTEL is compared with other three algorithms EXPEDITE, CHARM and CLOTEL manually by considering a very small set of itemset. Consider an example transaction database as in Table2.1, whose total number of closed frequent itemsets are 6 (acdf x 14 ; cef x 135 ; ae x 12 ; cf x 1345 ; a x 124 ; e x 1235) and also consider following another transaction table for analysis :

Table 1. Sample transaction dataset 1

Transaction ID	Items	Freq(X) [min_sup = 3 or 50%]
1	ACTW	ACTW
2	CDW	CDW
3	ACTW	ACTW
4	ACDW	ACDW
5	ACDTW	ACDTW
6	CDT	CDT

For the above table the total number of closed frequent itemsets are 7 (ACTW x 135; CDW x 245; ACW x 1345; CD x2456; CT x 1356; CW x 1245 ; C x 123456). For this two different kind of transaction table I had applied all the three algorithm manually, and there analysis is shown in Fig.2

For Table 1;

- CHARM & EXPEDITE requires 5 projection to find all 7 closed frequent itemsets.Its projection order is (D T A W C)
- CLOTEL needs 2 projection. Its projection order is (ACDTW). Its also has 3 dummy projection , which is shown in negative values.
- CHARM needs only 1 projection . Its projection order is (CWATD)

Table 2. Sample transaction dataset 2

Transaction ID	Items	Freq(X) [min_sup = 2 or 40%]
1	a, c, d, e, f	a, c, d, e, f
2	a, b, e	a, e
3	c, e, f	c, e, f
4	a, c, d, f	a, c, d, f
5	c, e, f	c, e, f

For Table 2,

- CHARM & EXPEDITE need four projection to determine all the 6 closed itemset , say first projection determine 1,second and third projection determine 2 and finally fourth projection determine 1 itemset. Its projection order (d a e c f).
- CLOTEL need four projection to determine all the 6 closed itemset, were 3rd projection is a dummy projection which is shown in negative values. Its projection order (a c d e f)
- CIMTEL required only three projection. Its projection order (c f a e d)

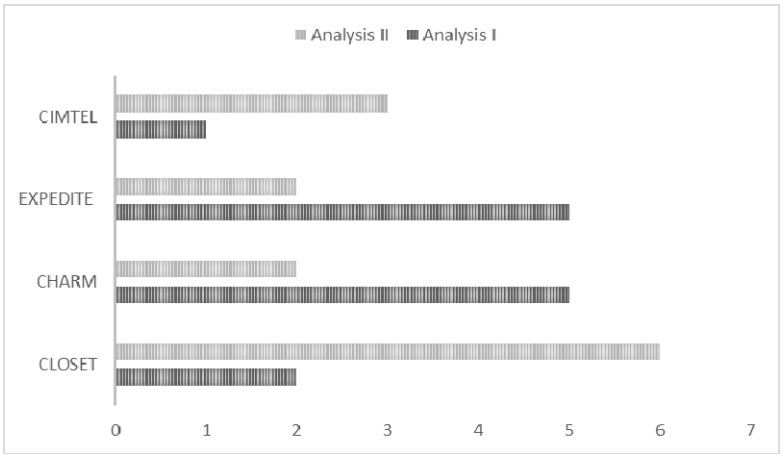

Figure 2 Manual Analysis

D. CONCLUSION

The computer and telecommunication industries rely heavily on knowledge-based expert systems to manage the performance of their networks. In this project we design a intelligent algorithm called CIMTEL to determine the closed frequent itemset , which would be helpful to apply any kind of Association Rule for knowledge acquisition process for determination of any fraud detection , network fault recovery, etc., This algorithm out perform the former algorithm namely CLOTEL,CHARM and EXPEDITE effectively by tremendously reducing the number of iteration spend for determining the complete closed frequent itemsets. The Future work of this work is to analyse the complexity in mining closed frequent itemset using CIMTEL, Extent the algorithm for biological database by including some pattern recognition technique.

REFERENCES

[1] Agarwal, Ramesh C., Charu C. Aggarwal, and V. V. V. Prasad. "A tree projection algorithm for generation of frequent item sets." Journal of parallel and Distributed Computing 61, no. 3 (2001): 350-371.

[2] Agrawal, Rakesh, and Ramakrishnan Srikant. "Fast algorithms for mining association rules." In Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, pp. 487-499. 1994.

[3] Daszczuk, Wiktor, Piotr Gawrysiak, Tomasz Gerszberg, Marzena Kryszkiewicz, Jerzy Mieścicki, Mieczysław Muraszkiewicz, Michał Okoniewski et al. "Data mining for technical operation of telecommunications companies: a case study." In Proceedings of International Conference SCI/ISAS. 2000. Han, J. & Kamber, M. (2001). Data mining: Concepts and techniques. Academic Press

[4] Lucchess, C., Orlando, S. & Perego, R. (2004). Mining frequent closed itemset without duplicates generation. Proceedings of 2004 ACM-SIGMOD International Conference on Management of Data.

[5] Pei, J., Han, J., & Mao, R. (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. Proceedings 2000 ACM-SIGMOD International Workshop Data Mining and Knowledge Discovery (DMKD'00).

[6] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. C. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. Proceedings 2001 International Conference Data Engineering (ICDE'01).
[7] Zaki, Mohammed Javeed, and Ching-Jiu Hsiao. "CHARM: An Efficient Algorithm for Closed Itemset Mining." In SDM, vol. 2, pp. 457-473. 2002. Guizhen Yang. The Complexity of Mining maximal Frequent Itemsets and Maximal Frequent Patterns ACM transaction on Database System 2004.
[8] Massimo Quadrana, Albert Bifet and Ricard Gavalda 2014"An efficient closed frequent itemset miner for the MOA stream mining system" AI Communications, vol.28 no.1,pp 143-158
[9] Laszlo Szathmary et al 2014 "A fast compound algorithm for mining generators, closed itemsets and computing links between equivalence classes" Aannals of Mathematics and Artificial Intelligence, vol. 70, no.2, pp. 81 - 105.
[10] Guiulio Aliberti et al "EXPEDITE: EXPress closedED ITemset Enumeration", Expert Systems with Applications, vo.42, no.8,2015, pp. 3933-3944
[11] Andras Kiraly et al "Novel techniques and an efficient algotithn for closed pattern mining" Expert Systems with Applications vol.41, no.11, 2014, pp 5015-5114.
[12] Pasquier, Nicolas, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. "Efficient mining of association rules using closed itemset lattices." Information systems24, no. 1 (1999): 25-46.
[13] Hatonen, Kimmo, Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, and Hannu Toivonen. "Knowledge discovery from telecommunication network alarm databases." In Data Engineering, 1996. Proceedings of the Twelfth International Conference on, pp. 115-122. IEEE, 1996.
[14] Wu, Pei-Hsin, Wen-Chih Peng, and Ming-Syan Chen. "Mining sequential alarm patterns in a telecommunication database." In Databases in Telecommunications II, pp. 37-51. Springer Berlin Heidelberg, 2001.
[15] Chen, Min, Shiwen Mao, Yin Zhang, and Victor CM Leung. "Big Data Analysis." In Big Data, pp. 51-58. Springer International Publishing, 2014.
[16] G.Nagarajan, K.K.Thyagharajan "Rule-Based Semantic Content Extraction in Image using Fuzzy Ontology" International Review on Computers and Software Vol.9,Issue.2,PP.266-277,2014
[17] G.Nagarajan, R.I.Minu "Fuzzy ontology based multimodal semantic information retrieval", Procedia Computer Science-Journal Elsevier , Netherlands, Vol.48, PP.101-106 ,2015.