International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 18 (2017) pp. 7783-7789

Parallel Frequent Dataset Mining and Feature Subset Selection for High Dimensional Data on Hadoop using Map-Reduce

Sandhya S Waghere

Research Scholar, Department of Computer Science and Engineering,
K L University, Green Fields, Vaddeswaram, Guntur District, Andhra Pradesh, India.
Assistant Professor in PCCOE, Nigdi, India.

Orcid Id: 0000-0002-2349-9639

Pothuraju Rajarajeswari

Professor, Department of Computer Science and Engineering,
K L University, Green Fields, Vaddeswaram, Guntur District, Andhra Pradesh, India.

Orcid Id: 0000-0002-2349-9639

Abstract

Data mining mostly use for information analysis and finding frequent dataset. Now a days cloud computing is used for information storage and many other data processes like data mining, data retrieval, data distribution etc. As data increasing very rapidly on server day by day, many complications are introduced. Most common problems are load balancing on server and time optimization. To overcome these limitations parallel frequent dataset mining is very effective method. Fidoop parallel frequent dataset mining algorithm which is based on mapreduce framework helps to improve load balancing and FiDoop-HD, speed up the mining performance for high-dimensional data analysis. Fidoop is very efficient and scalable algorithm for large clusters of data.We are using Fast Clustering Based Feature Selection Algorithm for High Dimensional Data which uses minimum spanning tree (MST) to divide data into different clusters and unfasten unrelated sets and gives accurate and efficient result with similar sets.

Keywords: frequent item sets, Frequent Items Ultrametric trees, Hadoop, MapReduce.

INTRODUCTION

Frequent itemset mining is an entrancing division of information mining that controls arrangements of activities or events. The first algorithm for mining frequent itemsets, which was created in 1993 by Agrawal is still frequently utilized. Essential idea of frequent itemset mining algorithm is first sweep the database to locate all frequent 1-itemsets, at that point continuing to locate all frequent 2-itemsets, at that point 3-itemsets and so forth. At every cycle, candidate itemsets of length n are produced by joining frequent itemsets of length n – 1; the frequency of every candidate itemset is assessed before being added to the arrangement of frequent itemsets.

Information mining is a one of strategy utilized for finding the example from the immense measure of information. There are numerous information mining algorithms are created like grouping, bunching, and affiliation run the show. The most conventional and basic algorithm is the affiliation decide that is partitioned into two sections I) producing the frequent itemset ii) creating affiliation govern from all itemsets. Frequent itemset mining (FIM) is the main issue in the affiliation manages mining. Consecutive FIM algorithm experiences execution weakening when it worked on gigantic measure of information on a solitary machine to address this issue parallel FIM algorithms were proposed.

The FIUT algorithm comprises of two stages. In the first stage examine a database in two rounds. The first examine creates frequent one itemsets by processing the help of all things, though the second sweep brings about n-itemsets by pruning every infrequent thing in every exchange record. Note that, n means the quantity of frequent things in an exchange. In stage two, a n-FIU-tree is over and over built by breaking down every h-itemset into n-itemsets, where n + 1 = h = (M is the maximal estimation of n), and unioning unique n-itemsets. At that point, stage two begins mining all frequent k-itemsetsbased on the leaves of n-FIU-tree without recursively navigating the tree. Contrasted and the FP-development technique, FIUT significantly diminishes the figuring time and storage room by turning away overhead of recursively looking and crossing restrictive FP trees.

To find FIM, two well-known algorithms are Apriori and FP Growth (Frequent Pattern). Apriori [7] uses the candidate generation approach and has to repeatedly scan the database. To reduce the time required for scanning of database and

without generating candidate itemsets, the next approach is FP Growth. But FP-growth approach has infeasibility to construct in-memory trees. To develop and address these problems of existing algorithm parallel FIM is proposed using MapReduce programming paradigm. Use of MapReduce programming model to solve the above problem which can handle large datasets through number of clusters. This distributed approach is combined with FIM to overcome the drawbacks of sequential FIM and to increase performance [3]. So the approach is called as Fidoop as it uses Hadoop- MapReduce model to find frequent itemsets. In this approach FIUT (Frequent Items Ultrametric Tree) algorithm used to provide compressed storage, to avoid conditional pattern bases, to reduce I/O overhead [3]. In FIUT no need to traverse entire tree only to check leaf nodes as frequent itemsets.

Using MapReduce model it is easy to handle the large datasets across number of clusters. This distributed method is combined with FIM to overcome the drawbacks of sequential FIM and hence performance gets increased. Instead of FIUT with mapreduce, an enhanced Apriori algorithm (EA) is used to reduce number of scans on datasets. It mainly focuses on data partitioning method and load balancing of datasets. It takes less time compared to old apriori algorithm. Mappers and reducers works in parallel to boost speed and fine load balance among clusters.

LITERATURE SURVEY

Fast Parallel Mining of Frequent Itemsets (H. D. K. Moonesinghe, Moon-Jung Chung, Pang-Ning Tan): A parallel approach was applied to the Frequent Pattern Tree (FP-Tree) algorithm, which is a fast and popular tree projection based mining algorithm. This approach carries out the mining task parallel until all the frequent patterns are generated and build several local frequent pattern trees. Fast parallel mining method achieved good workload balancing among processors at runtime researcher by developing a dynamic task scheduling strategy at different stages of the algorithm. Experimental results of system showed parallel algorithm resulted in higher speedups in almost all the cases compared to the sequential algorithm. Also, parallel algorithm showed scalable performance for larger data sets[9].

Y.-J. Tsay et al.[2], proposed a "FIUT: A new technique for mining frequent itemsets,". This paper proposes a dynamic technique, the frequent itemset ultra metric trees (FIUT), for mining continuous itemsets in a database. FIUT exploits an unusual incessant things ultrametric tree (FIU-tree) structure to improve its ability in getting incessant itemsets. FIUT has four remarkable advantages. First one, it minimizes I/O overhead by inspecting the database just twice. Next one is, the FIU-tree is animproved method to segment a database which results from combination exchanges and fundamentally reduces theinquiry space. Other one, just continual things in each exchange are embedded into the FIU-tree for completely packed storage. Last one, all successive itemsets are produced by examine the leaves of each FIU-tree, without overpassing the tree recursively, which altogether reduced processing time.

E.-H. Han, G. Karypis, and V. Kumar[3],depicts "Scalable parallel data mining for association rules,". One of the important issues in information mining is finding association rules from databases of relations where every exchange includes of a set of items. The most time demolishing operation in this disclosure procedure is the calculation of the recurrence of the events of applicants in the database of exchanges. To prune the exponentially wide space of applicants, peak existing calculations, consider just those applicants that have a client characterized least backing.

K.-M. Yu et al.[4],presented "A load-balanced distributed parallel mining algorithm," Due to the exponential advancement in general information, associations requireto deal with a never-endingly creating measure of cutting edge information. A strongest among the most fundamental challenges for data mining is quickly and adequately finding the relationship among data. The Apriori algorithm has been the most understood technique in finding persistent illustrations. In any case, while applying this methodology, a database must be checked various circumstances to figure the checks of endless itemsets. Parallel and scattered calculations is suitable for reviving the mining procedure. In this paper, the Distributed Parallel Apriori (DPA) estimation is proposed as a response for this issue. In this reference, metadata are secured as Transaction Identifiers (TIDs), with the end goal that only a singular range to the database is required. The approach in like manner takes the component of itemsetcounts into thought, thusly creating a balanced workload among processors and reducing processor unmoving time. Tests a PC bundle with 16 handling centersis moreover made to show the execution of this approach and complexity it and some other parallel mining calculations. The test outcomes show that the proposed approach beats the others, especially while the base sponsorships are low.

L.Zhou et al.[5], presented "Balanced parallel FP-growth with MapReduce", General itemset mining (FIM) accept a key part in mining affiliations, associations and various other basic data mining errands. Lamentably, as the volume of dataset gets greater well ordered, the majority of the FIM figuring's in composing get the opportunity to be incapable as a result of either unreasonably enormous resource essential or too much correspondence cost. In this reference, it propose a balanced parallel FPGrowth count BPFP, in light of the PFP estimation [1], which parallelizes FPGrowth in the MapReduce approach. BPFP incorporates into PFP stack equality highlight, which upgrades parallelization and thusly improves execution. Through correct examination, BPFP beat the PFP which uses some clear assembling system.

Frequent item set mining used in vast kind of application areas such as decision support system, web usage mining, bioinformatics, network analysis, sentiment analysis, etc. There are variety of algorithms proposed by different researchers for frequent Itemset Mining. Each of it has its own advantages and disadvantages. Following is the review of some of the research papers from various conferences and publications.

Association rule mining is method for finding remarkable relations among itemsets in large datasets. [1][16]. It is for identifying strong rules founded in databases using some criteria. It is presented with issues of digging out the frequent items from very huge database. It comes with rules that have minimum transactional support and minimum confidence. The algorithm estimates the itemsets for one pass. Similarly it adjusts among the number of passes over data and itemsets that are calculated in a pass. To avoid certain itemsets pruning is done. Thus gives factual association itemsets from large databases. Advantages of this algorithm is it uses buffer management method which is not fit in the memory in one pass and so will move to next pass.

For Association Rule Mining in MapReduce [19] PARMA: A Parallel Randomized Algorithm is proposed. It is parallel Randomized method for finding out frequent itemsets in big datasets. It is beneficial for association rules in data mining. PARMA works with two tasks first collects the random data samples and another applies parallel processing method to improve the speed of mining. This procedure excludes the replication which is very costly. PARMA works by making many random fragments of the data exchanges and then applies mining process then for parallelization mapreduce used. Thus performance improved than earlier algorithms, it gives frequent itemset

The proposed research work presents the problem related with extraction of frequent items from huge datasets. It presents rules that have minimum transactional support and minimum confidence. The proposed algorithm that calculates the itemsets for one pass and adjust among the number of passes over data and itemsets. This calculation uses cropping technique to avoid certain itemsets. Thus gives right related itemsets from large databases. Benefit of this algorithm is, buffer management technique which is not fit in the memory in one pass and so it shifts to next pass.

To investigate applicability of FIM techniques on the MapReduce platform. For this they have used two algorithms i) Dist-Eclat and ii) BigFIM.In the first technique that is Dist-Eclat, it distributes the search space equally as possible among mapper.

Algorithm operates in three steps: by using vertical database rather than transaction database, in the first step the vertical database is divided into equal sized blocks called shards and distributed to available mappers. Each mapper extracts the frequent singletons from each block and gives to the reducer. The reducer collects all the frequent tested. In the second step the set of frequent itemsets of size K are generated (Pk). Frequent singleton itemsets are distributed to the mappers. Each mapper runs Éclat to find frequent K-sized superset of items. The reducer collects all the frequent K-sized supersets of items and distributes it to the next batch of mappers. Round Robin approach is used for the distribution of the frequent itemset. The third step is the mining the prefix tree. The mutual information between the mappers are independent, so mapper complete each step independently.

Advantage of this algorithm is it focuses on speed thus provide efficient in terms of speed. One of the disadvantage of this algorithm is that this technique mines large dataset but not massive datasets.

Second algorithm is the BigFIM which overcomes the problem of DistEclat. It also works in three steps: in first step K-FI's are generated using breath-first method.

Every mapper takes the database and gives itemsets for which, support can be calculated. The reducer takes all itemsets and returns only the global frequent itemsets. These itemsets are considered as candidates and distributed to the mappers for breath-first search. This process continues K-times to generate K-FI's. Next it computes the possible extension. The mapper gives local Tid-list to the reducer, then reducer combines the local Tid-lists, to one Tid-list, and assigns prefix to mappers. In the final step, mapper works on individual prefix group.

Advantage of this algorithm is, it is used to be optimized to run on large datasets, thus provides scalability. Limitation of the approach is that it has workload distribution issues.

Parallel FP Growth algorithm using balanced partitioning (BPFP) [5]-[9] is another important proposed system for pattern mining. It works in two stages, in first stage of BPFP, based on conditional pattern load is computed. In next stage, this load is get divided into many groups. For this purpose, MapReduce paradigm is used on FP algorithm.Benefit of this approach is, it enhances parallelization and so improves execution. But Lacks in automatic parallelization.

To overcome the drawbacks of both Apriori and FP Growth algorithm, an effective method, the frequent items ultrametric trees (FIUT), for mining frequent itemsets in a database is introduced [2]. It's a tree (FIU-tree) structure to improve efficiency in finding frequent itemsets.As compared to other existing techniques, FIUT has some benefits. i) It reduces I/O overhead by scanning the database only two times. ii) To partition a database the FIU-tree is a better way, which results from clustering transactions, and reduces the search space. iii) Only frequent items in each transaction are inserted as nodes into the FIU-tree for compacted storage. iv) Frequent itemsets are generated by checking the leaves of each FIU-tree, no need to traverse tree recursively, which reduces computing time.

The working of FIUT is as follow: It consists of two phases- Phase 1: Two rounds of scanning of database. The first scan generates frequent one-itemsets through computing the support of all items, in second scan k-itemsets are generated by trimming the infrequent itemsets. k-denotes the number of frequent items in a transaction. Phase 2- A k-FIU tree is repeatedly built by decomposing each h-itemset into k-itemsets, where $k + 1 \leq h \leq M$ (M is maximal value of k), and combining original k-itemsets. Then phase two starts mining all frequent k-itemsets based on the leaves of k-FIU tree without recursively traversing the tree. Compared to FP growth method, FIUT significantly reduces the computing time and storage space by avoiding overhead of recursively searching and traversing conditional FP tree. Demerit of this algorithm is sometime it omits few itemsets.

Following approach introduces with how to achieve scalability and robustness to handle huge datasets, to mine relations, patterns and to make business decisions using quicker and efficient parallel processing strategy- MapReduce [2]. The method ClustBigFIM gives mix approach for frequent itemset mining for large data sets using combination of parallel k-means, Apriori algorithm and Eclat algorithm; by increasing scalability and performance which overcomes limitation of Big FIM.

Proposed technique ClustBigFIM works with these four steps i) Find Clusters uses parallel k-means algorithm for generating clusters. ii) Finding k-FIs, uses apriori algorithm for finding frequent itemsets. iii) Generate Single Global TID list and iv) Mining of Subtree.

One of the advantage of this algorithm is it provides speed and scalability but sometimes provides approximate results.

The remarkable system that gives the parallel mining of frequent itemsets using Hadoop-MapReduce programming paradigm which not only gives parallelization but also handles load balancing, fault tolerance, and data distribution on huge cluster [5]. The approach is called as Fidoop. To get compressed storage and avoid conditional pattern bases, fidoop integrates FIUT structure than FP trees. There are three Mapreduce jobs who plays important steps in entire frequent itemset mining approach.

The first MapReduce job creates one-itemsets. Transactional database divided into many input files stored by HDFS on data nodes of Hadoop clusters. Each mapper serially reads the input and compute the frequencies of items and create one-itemsets. Reducer merges and sorts the mapper's one-itemsets, in the form of output pair <Text item, LongWritable count>, and stored in local file named as F-list, which will become input for second Mapreduce job.

The second MapReduce job scans the database and with reference list of one-itemset from first MapReduce job, cut the infrequent items from each transaction record. And provide k-itemsets. The output of reducer is key/value pair with key as the number of each itemset and value is each itemset and its count.

The third MapReduce job is difficult one as it does i) Decomposing itemsets, ii) building k-FIU tress and Mining of frequent itemsets. Third job is more scalable as decomposing of each mapper in not dependent on other mappers. The output of the Map function is in the form of key value pairs where keys are number of items in itemsets and value are FIU tree with leaf and nonleaf nodes. The reducer builds the k2-FIU tree and mines the frequent itemsets just by examining count of each leaf in k2FIU tree without repetitively navigating tree.

Advantages of this technique are i) Use of many mappers to decompose h-itemsets in parallel increases data storage efficiency and I/O performance. ii) Provides scalability and better load balancing. But the approach is works well with homogeneous clusters not with heterogeneous clusters.Differant methods used in parallel mining of frequent itemsets. Following table shows comparative study of different methods [9].

Table 1: Methods used in parallel mining of frequent item sets.

Title	Method	Description
1.Tree partition based parallel frequent pattern mining on shared memory systems	shared memory parallelization of FP-Growth algorithm	It lacks in load balancing, need large database for storage, cannot be used on large database
2. FIUT: A new method for mining frequent itemsets"	Improved FIU-tree .It is an improved method to partition a database by clustering the transactions and significantly reducing the search space	Lacks in automatic arallelization And load balancing
3. Scalable parallel data mining for association rules	Distribution of data over data nodes, parallel mining	Not efficient, require more time for mining
4 A load balanced Distributed parallel mining algorithm	Parallel mining on database, work load balance	Lacks in automatic parallelization And is expensive data mining
5. Balanced parallel FP-growth with MapReduce"	Parallel FP growth, use MapReduce programming model for large database	Lacks in load balancing and is expensive illuminate the adaptability and burden adjusting mining

EXISTING SYSTEM

Existing system do well to find frequent itemsets in parallel maaner using mining algorithm called FiDoop through the MapReduce programming model. By using frequent ultarmetric trees (FIUT) it is possible to achieve compressed storage than using traditional FP trees. The working of fidoop is, it works using three map reduce jobs for mining. The first mapreduce process find frequencies of each item by scanning the dataset first time. In second mapreduce process it provides the k-itemsets with the help of frequency count of first mapreduce job and scanning dataset second time. In last third mapreduce job, map process separately decomposes itemsets and reducers do the combination by making small ultrametric trees, and mine these trees individually.

Existing System Positive Mark

Existing system performs well for parallelization. Also performance is increased by balancing I/O load among data nodes of clusters. Its three mapreduce phases useful for distribution of large data.

Existing System Negative Mark

One of the drawback of existing system is, for given large datasets the method for partitioning of data is suffer from I/O and mining efforts due to excessive transactions transmitted between nodes. Second, FIUT algorithm gives data leakage problem and reduces performance.

Proposed system overview

Using mapreduce paradigm it is possible to implement frequent itemset mining algorithm which gives parallelism. Thus it is possible to reduce drawbacks of traditional system and get automated parallelization, load balancing and efficient data distribution. Proposed system uses Enhanced Apriori algorithm (EA) to solve the problems of FIUT algorithm. Through this method is it is possible to reduce time that is required for scanning the transactions.

FAST Clustering Algorithm

Clustering is the way toward gathering an arrangement of physical or dynamic items into classes of comparable articles. It is a normal and imperative undertaking that finds numerous applications in IR and different spots. The execution, heartiness, and handiness of clustering calculations are relies upon Finding likenesses between information as indicated by the qualities found in the information and gathering comparative information objects into groups. The nature of a clustering result relies upon both the similitude measure utilized by the technique and its usage. So to increase quality and precision Quick clustering subset determination calculation is utilized. Feature determination technique is general type of feature extractions. In feature extraction, new feature set is produced from the information features which are now preset. While in feature determination process, it gives the subsets of feature which are valuable for required hunt. It gives points of interest counting less time for looking, ideal outcomes and so forth. The feature subset choice should be possible with the assistance of different calculations, for example, best pursuit, greedy forward choice calculation, greedy in reverse disposal calculation, hereditary calculation [10]. Many feature subset choice calculations have been proposed for machine learning applications. They can be classified into four classifications: the Embedded, Filter, Wrapper, and Hybrid methodologies [10]. The wrapper strategy used to decide the decency of the chose subsets, the precision of this calculation is commonly high. Be that as it may, the computational multifaceted nature is vast. The channel techniques are autonomous of learning calculations, with great all inclusive statement. Their computational multifaceted nature is low, however the exactness of the calculations is not ensured [10]-[21]-[22].

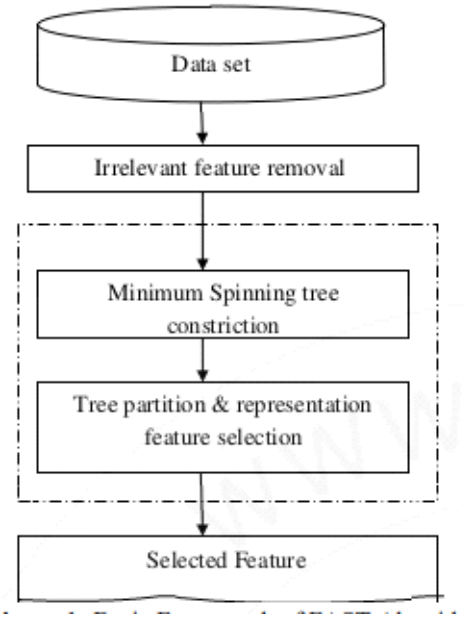

Figure 1: FAST Algorithm Framework

The wrapper techniques are computationally costly and not valuable on little preparing sets [10]-[21]. The channel strategies are typically a decent decision when the quantity of features is exceptionally expansive i.e. for high dimensional information.

Quick calculation is a powerful approach to diminish measurements, evaluating unimportant information and to create result in high successful way. To accomplish objective

of FAST clustering calculation, it works in two stages. In the initial step, it partitions information into groups utilizing chart strategies. For this reason Minimum Spanning Method is utilized. In second step of FAST, the subsets that are most precise or relative with the required inquiry are chosen from group and frame a feature subset [10].

Feature subset distinguishes and expels the same number of insignificant and repetitive features as would be prudent. As we are clustering all the features with their relations, all the related feature set and inconsequential feature set are grouped independently. This clustering deals with related and random feature sets for forecast according to the required target look. Albeit a few of existing framework or calculation has capacity to expel the disconnected feature set, some don't includes effectiveness and adequacy to accomplish objective. Embraced Minimum Spanning Tree Computes an area chart of in-positions, at that point erase any edge in the diagram that is any longer or shorter (as per some basis) than its neighbors. The result will be a backwoods and each tree in those woods speaks to a bunch of feature. This group is utilized to feature subset choice.

Negative mark

Apriori calculation needs to check the whole database over and again.

CONCLUSION

Analysis of big datasets gives predictions and valuable information that can be used for many applications such as business decisions, sentiment analysis, market basket analysis, web logs analysis, drug design(molecular fragment mining) and likewise. There are various techniques and algorithms available for frequent itemsets mining. For solving scalability and load balancing issues in traditional parallel mining algorithms new approach is introduced with the use of big data analysis tool Hadoop- MapReduce. With the help of MapReduce paradigm it is possible to find frequent itemsets in parallel more effectively. Fidoop is the approach which uses MapReduce programming model to produce frequent itesets with handling the issues related to load balancing, fault tolerance and data distribution on large size clusters.

The proposed system under implementation uses enhanced apriori (EA) algorithm to overcome the drawbacks of FIUT algorithm. The system will work with three mapreduce jobs. The first mapreduce job generates all 1-itemsts. Second mapreduce job gives k-itemsets by pruning infrequent itemsets. The enhanced apriori to be implemented on third phase of mapreuduce job.

REFERENCES

[1] D. Chen et al., “Tree partition based parallel frequent pattern mining on shared memory systems,” in Proc. 20th IEEE Int. Parallel Distrib. Process. Symp.(IPDPS), Rhodes Island, Greece, 2006, pp. 1–8.

[2] Y.-J. Tsay, T.-J. Hsu, and J.-R. Yu, “FIUT: A newmethod for mining frequent itemsets,” Inf. Sci., vol. 179, no. 11, pp. 1724–1737, 2009.

[3] E.-H. Han, G. Karypis, and V. Kumar, “Scalableparallel data mining for association rules,” IEEE Trans. Knowl. Data Eng., vol. 12, no. 3, pp. 337–352,May/Jun. 2000.

[4] K.-M. Yu, J. Zhou, T.-P. Hong, and J.-L. Zhou, “A load-balanced Distributed parallel mining algorithm,” Expert Syst. Appl., vol. 37, no. 3, pp. 2459–2464, 2010.

[5] L. Zhou et al., “Balanced parallel FP-growth withMapReduce,” in Proc. IEEE Youth Conf. Inf. Comput. Telecommun. (YC-ICT), Beijing, China, 2010, pp.243–246.

[6] “ECLAT Algorithm for Frequent Itemsets Generation”, ManjitkaurUrvashi Grag Computer Science and Technology, Lovely Professional University Phagwara, Punjab, India . InternationalJournal of Computer Systems (ISSN: 2394-1065), Volume 01– Issue 03, December, 2014 Available at http://www.ijcsonline.com/

[7] “Implementation Of Parallel Apriori Algorithm On Hadoop Cluster”, A. Ezhilvathani1, Dr. K. Raja. International Journal of ComputerScience and Mobile Computing.

[8] M. Chen, X. Gao and H. Li, "An efficient parallel FP-Growth algorithm," 2009 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Zhangijajie, 2009,pp.283-286.DOI: 10.1109/CYBERC.2009.5342148

[9] Mrs. Sandhya S. Waghere, Dr.Pothuraju Rajarajeswari, “Survey on Achieving Best Knowledge from Frequent Item set Mining using Fidoop”, International Journal of Computer Applications (0975 – 8887) Volume 171 – No. 9, August 2017,

[10] Qinbao Song, Jingjie Ni and Guangtao Wang, “A Fast clustering based feature subset selection algorithm for high dimensional data”, In proceedings of the IEEE Transactions n Knowledge and data engineering, 2013.

[11] L. Yu and H. Liu, “Feature Selection for HighDimensional Data: A Fast Correlation-Based Filter Solution,” Proc. 20th Int’l Conf. Machine Leaning, vol. 20, no. 2, pp. 856-863, 2003.

[12] M. Dash, H. Liu, and H. Motoda, “Consistency Based Feature Selection,” Proc. Fourth Pacific Asia Conf.

Knowledge Discovery and Data Mining, pp. 98-109, 2000.

[13] R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Rec., vol. 22,no. 2, pp. 207–216, 1993.

[14] Y. Lin, P.-Y. Lee, and S.-C. Hsueh, "Apriori-based frequent itemset mining algorithms on MapReduce", in Proc. 6th Int. Conf. Ubiquit. Inf. Manage. Commun. (ICUIMC), Danang, Vietnam, 2012.

[15] YalingXun, Jifu Zhang and Xian Qin, "Fidoop: Parallel mining of frequent Itemsets using MapReduce", IEEE Trans.on sys.man and cybernetics, Vol. 46,No.3, March 2016

[16] Sandy Moens,Emin Aksehirli and Bart Goethals, " "Frequent Itemset Mining for BigData", intl. conf on Bigdata", IEEE 2013.

[17] Sheela gole and Bharat Tidke, "Frequent Itemset Mining for BigData in social media using ClustBigFIM algorithm", Intl Conf.on Pervasive Computing, IEEE 2015.

[18] Kiran Chavan, Priyanka Kulkarni, Pooja Ghodekar, S. N. Patil, "Frequent itemset mining for Big data ", IEEE,Green Computing and Internet of Things (ICGCIoT), 2015.

[19] M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal, "PARMA:A parallel randomized algorithm for approximate association rules mining in MapReduce," in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.,Maui, HI, USA, 2012.

[20] Wei Lu,Yanyan Shen,Su Chen,Beng Chin Ooi, "Efficient Processing of k Nearest Neighbor Joins using MapReduce"2012.

[21] R.Munieswari,"A Survey on Feature Selection Using FAST Approach to Reduce High Dimensional Data",IJETT, Volume 8 Number 5- Feb 2014.

[22] Jesna Jose,"Fast for Feature Subset Selection Over Dataset" International Journal of Science.