ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Mining Frequent Itemset Using Hadoop Framework

Sonal Rajabhau Londhe, Prof. Varsha R. Dange

Student, Dept of Computer, Dhole Patil College of Engineering, Savitribai Phule Pune University, Pune, India

Guide, Dept of Computer, Dhole Patil College of Engineering, Savitribai Phule Pune University, Pune, India

ABSTRACT: Data mining is the extraction of hidden predictive information from large databases, which is a powerful new technology with great potential to help companies as well as research focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. Frequent Itemset Mining is one of the classical data mining problems in most of the data mining applications. It requires very large computations and I/O traffic capacity. Also resources like single processor's memory and CPU are very limited, which degrades the performance of algorithm. In this paper we have proposed one such distributed algorithm which will run on Hadoop – one of the recent most popular distributed frameworks which mainly focus on Mapreduce paradigm. The proposed approach takes into account inherent characteristics of the Apriori algorithm related to the frequent itemset generation and through a block-based partitioning uses a dynamic workload management. The algorithm greatly enhances the performance and achieves high scalability compared to the existing distributed Apriori based approaches. Proposed algorithm is implemented and tested on large scale datasets distributed over a cluster.

KEYWORDS: Frequent Itemset Mining ,Mapreduce , HDFS

I. INTRODUCTION

Frequent itemset mining is a significant research subject in characterization, groupings and other fundamental data mining works. To discover frequent item sets is one of the essential computational assignments in association rule mining. As per the Association rule mining approach the frequent itemset is the itemset whose minimum support is greater than the given threshold value. These standard rules are useful for finding similar connections in the datasets and offers knowledge to the technique that produced the data. Presently, there arenumerous information from different sources like IT ventures, administrations, advances what's more, information. This vast information is available with different structures. To deal with such extreme information is exceptionally troublesome in light of the fact that it has billions of exchanges of clients, items and so on. There are number of techniques to get frequent itemsets from database. These strategies works on regular datasets, however not appropriate for massive information. To utilize frequent itemset mining strategy on extreme database is very hard process. To speed up the procedure of FIM is unpredictable and complex, because FIM expends huge time for calculation and high input / output intensity. One of the answers for this issue is to utilize another parallel frequent itemsets mining approach with MapReduce called Modified Apriori. Nowadays datasets are becoming very large and hence only sequential algorithms are not able to find the frequent itemsets correctly. Their performance gets degraded because of vast data. Therefore here we have used a Modified Apriori technique with the hash tables and MapReduce paradigm. MapReduce can tackle these issues with extensive number of databases crosswise over number of clusters. MapReduce will solve the parallel mining issue of large database and Modified Apriori algorithm will focus on the speed. This method improves the capacity of storage and computation of problem.

Modified Apriori algorithm utilizes the procedure of hash tables in which it stores past values to compare with new itemsets. It utilizes buckets in hash tables for storing results. Each buckets stores frequent items and will give a unique name, hence no need to scan repeatedly. FIUT is replaced with Modified Apriori. In this strategy, issues will be illuminated that are available in the current framework like data partitioning, load balancing of datasets. The working of mappers and reducers is done simultaneously to streamline the speed, well adjusting the load over different clusters.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

II. RELATED WORK

The authors of "Association Rule mining extracting frequent itemsets from the large database" have presented a problem of finding the frequent items from excessive database. The authors have developed the rules that have minimum transactional support and minimum confidence. For this an algorithm is used that carefully estimates the itemsets for one pass. It adjusts the data between the number of passes and itemsets that are measured in a pass. This process uses pruning system for avoiding certain itemsets. Hence this gives exact frequent itemsets from excessive databases [1]. Number of parallelization procedures is used to increase the performance of Apriori-like algorithms to find frequent itemsets. MapReduce has not only created but also exceeds in the mining of datasets of gigabyte scale or greater in either homogeneous or heterogeneous groups. Mapping administration of Mapreduce approach will minimize the overhead of each Mapreduce phase. The authors have implemented three algorithms, DPC, FPC, and SPC [2]. SPC has straight-forward functions and the FPC has static passes merged checking capacities. DPC consolidates the dataset of various lengths by utilizing dynamic strategy and it gives good performance over the other two calculations. Accordingly,these three calculations will scale up for the expanded dataset[2].When dataset gets larger the mining algorithms becomes inefficient to deal with such excessive databases. The authors have presented a balanced parallel FP-Growth algorithm BPFP [3], a revised version of PFP algorithm [1]. FP- growth algorithm is utilized with the MapReduce approach named as Parallel FP-growth algorithm. BPFP balances the load in PFP, which boosts parallelization and automatically enhances execution. BPFP gives a good performance by utilizing PFP's grouping system [3].

FIUT suggests a new technique for mining frequent itemsets called as Frequent Itemset Ultrametric Tree(FIUT)[4]. It is a sequential algorithm. It consists two main stages to scan the database. First stage calculates the support count for all itemsets in a large database. Second stage uses pruning method and gives only frequent itemsets. While calculating frequent one itemsets, stage two will construct small ultrametric trees. These results will be shown by constructing small ultrametric trees [4].

Dist-Eclat, BigFIM are two FIM algorithms used with MapReduce Framework. Dist-Eclat focuses on speed by load balancing procedure using k-FIS. BigFIM concentrates on hybrid approach for mining excessive data[5]. Apriori algorithm is additionally used to create k th FIS itemsets. The K th FIS is used to search frequent itemsets based on theEclat system. These three algorithms are used with round robin technique which achieves a better data distribution [5].

PARMA uses parallel mining approach with the benefits of Randomization for extracting frequent itemsets from vast amount of databases. This divides its functionality into two parts, firstly it gathers the arbitrary data samples and secondly it uses parallel computing method that is utilized to increase the mining speed. This method avoids the replication that is very expensive. This is done by making number of little arbitrary segments of the transactions. After that a mining algorithm applies on every segment individually. It applies parallel approach by utilizing MapReduce programming paradigm[6]. In the processing of k -Nearest Neighbor Joins utilizes MapReduce and distributes the excessive information on the number of machines. This is done by the mappers and the reducers give the results interms of the KNN join. KNN Join is the key component to search the kth-nearest neighbor. Mapreduce is utilized for effective computing the data to obtain the best performance result [7].

To diagnose the Heterogeneous Hadoop Cluster and to search primary faults, this paper is used Hadoop schedulers to produce efficient Hadoop clusters even if they are in heterogeneous clusters[8]. To differentiate and extract frequent and infrequent itemsets from the massive database, two phase of scanning will be done here. In first scan it accepts input and distributes it into mappers and finds out infrequent itemsets using minimum support. The reducer gives combined result and sends it to second phase. In this phase it scans first phase output and gives the final result[9].

FIUT is used with MapReduce to find frequent itemsets. MapReduce is apopular programming approach used to compting massive datasets [10]. It devides into three MapReduce phases. The first mapreduce phase finds out frequent one itemset. Here the database is divided into number of input files and given to each mapper. Second MapReduce phase scans the frequent one itemsets and prunes the infrequent itemsets. This phase generates k- frequent itemsets . Third mapreduce phase uses FIUT algorithm. This phase decomposes the itemsets and then it will create ultrametric tree[10]. FiDoop-DP-Data Partitioning uses the Map Reduce programming and gives the effective data partitioning technique for frequent itemset mining. FiDoop DP increases the performance by using the data partitioning technique, which is based on the Voronoi diagram. It extracts the correlations between the transactions. By consolidating the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

similarity and the Locality-Sensitive Hashing strategy, FiDoop-DP puts most similar records in data partition to increase locality and this is done without repeating records[15][16].
Parallel frequent mining for shared memory nodes can be done using tree division method. Tree-segment algorithm is used with parallel mining to extract frequent itemsets. The basic principle is to build a single FP-Tree in the memory, divide it into a few free parts and appropriate them to diverse strings. This algorithm cannot just diminish the impact of locks on the tree-building stage,also it reduces the input/output overhead that is harmful to the mining stage[16].

III. PROBLEM STATEMENT

The proposed work first investigate problem of Frequent Itemset Ultra-metric Tree (FIUT). System also focus on database security like SQL injection as well data collusion attacks etc and find efficient way for security as well execution in HDFS framework.

IV. SYSTEM ARCHITECTURE

In Proposed System a new data partitioning method to well balance computing load among the cluster nodes is developed, to meet the needs of high-dimensional data processing.

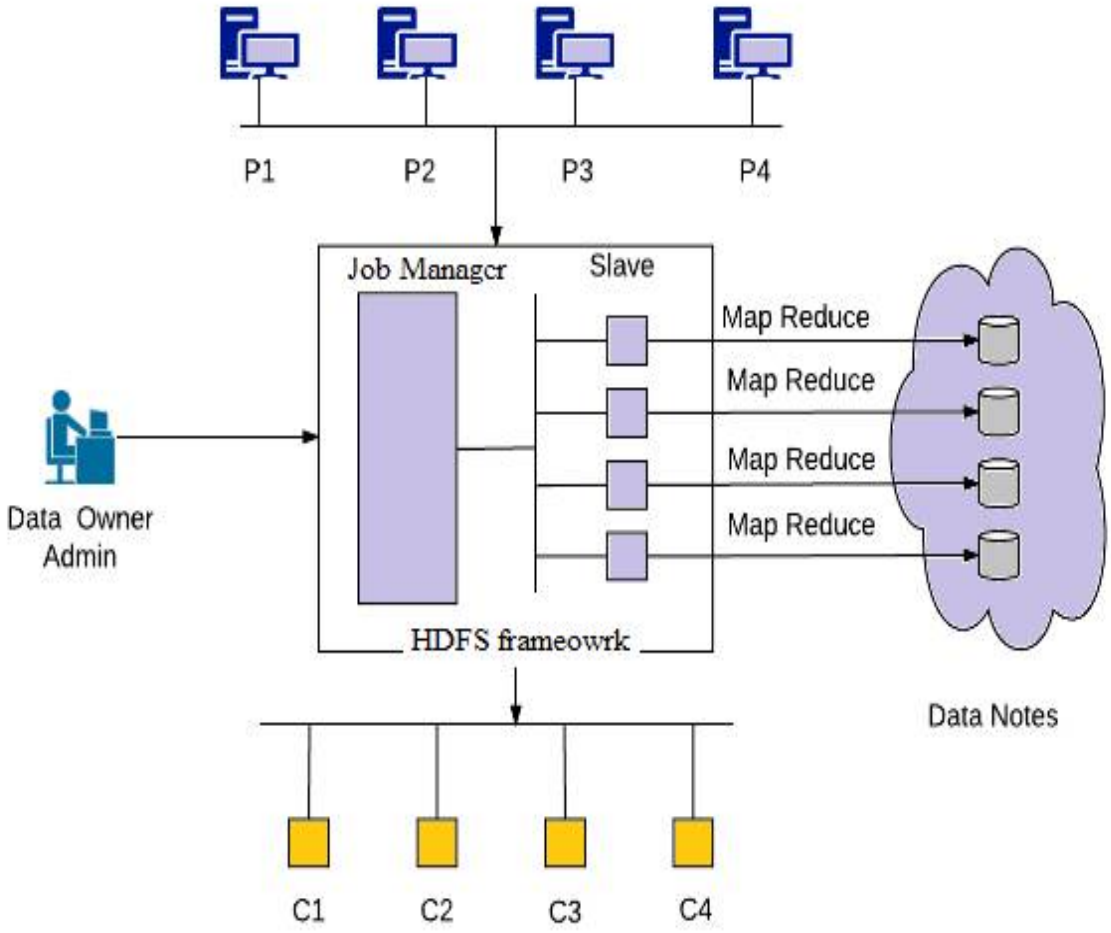


Figure 1: Proposed System Architecture

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Proposed Algorithm
Algorithm for SQL injection and Prevention
Phase : 1

```
1: Procedure SPMA(Query, SPL[ ])
INPUT: Query=User Generated Query
SPL[ ]=Static Pattern List with m Anomaly Pattern
2: For j = 0 to m do
3: If (AC (Query, String.Length(Query), SPL[j][0]) = =0 )then
4: Calc anomaly score
5: If ( ) Score Value Anomaly = Threshold
6: then
7: Return Alarm .. Administrator
8: Else
9: Return Query .. Accepted
10: End If
11: Else
12: Return Query .. Rejected
13: End If
14: End For
End Procedure
```

Phase : 2

```
1: Procedure AC(y,n,q0)
INPUT: y= array of m bytes representing the text input(SQL Query Statement)
        n= integer representing the text length(SQL Query Length)
q0=initial state (first character in pattern)
2: State : q0
3: For i = 1 to ndo
4: While g ( State, y[i] = = fail) do
5: State ← f (State)
6: End While
7: State ← g(State,.y[i])
8: If o(State)== NULL then
9: Output i
10: Else
11: Output
12: End If
13: End for
14: End Procedure
```

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

IV. RESULTS

File Upload

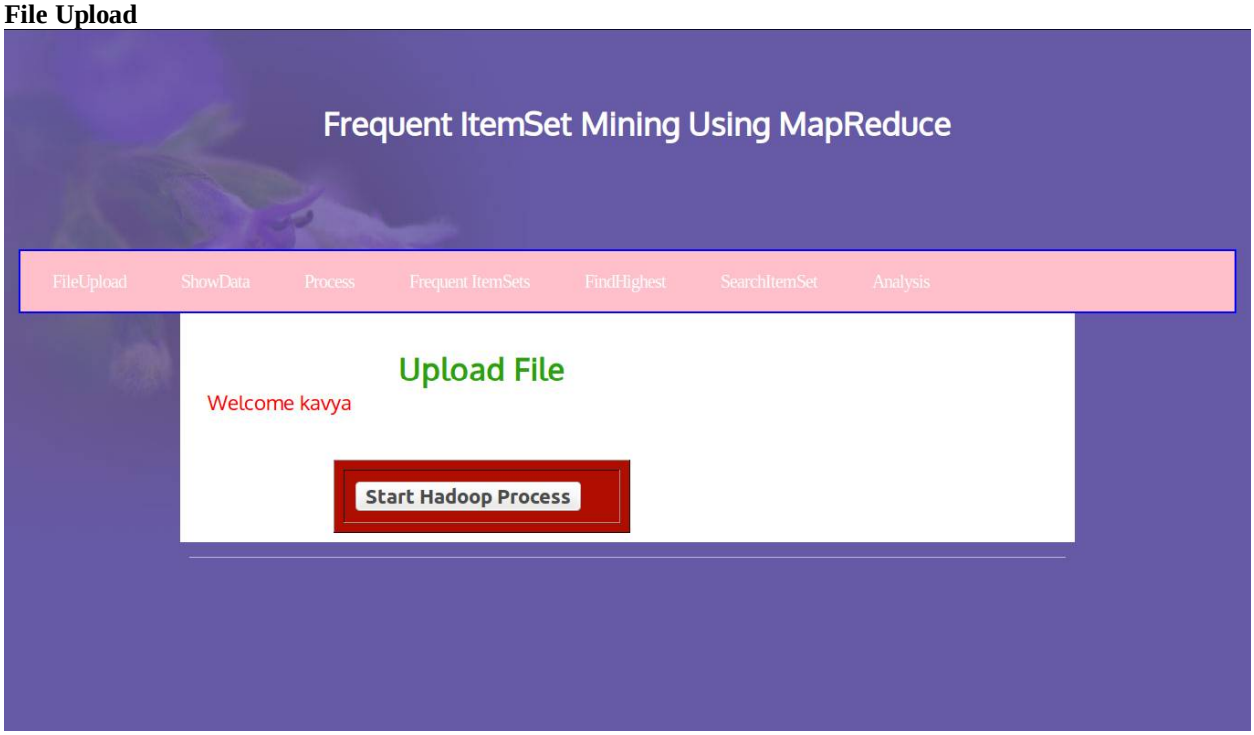
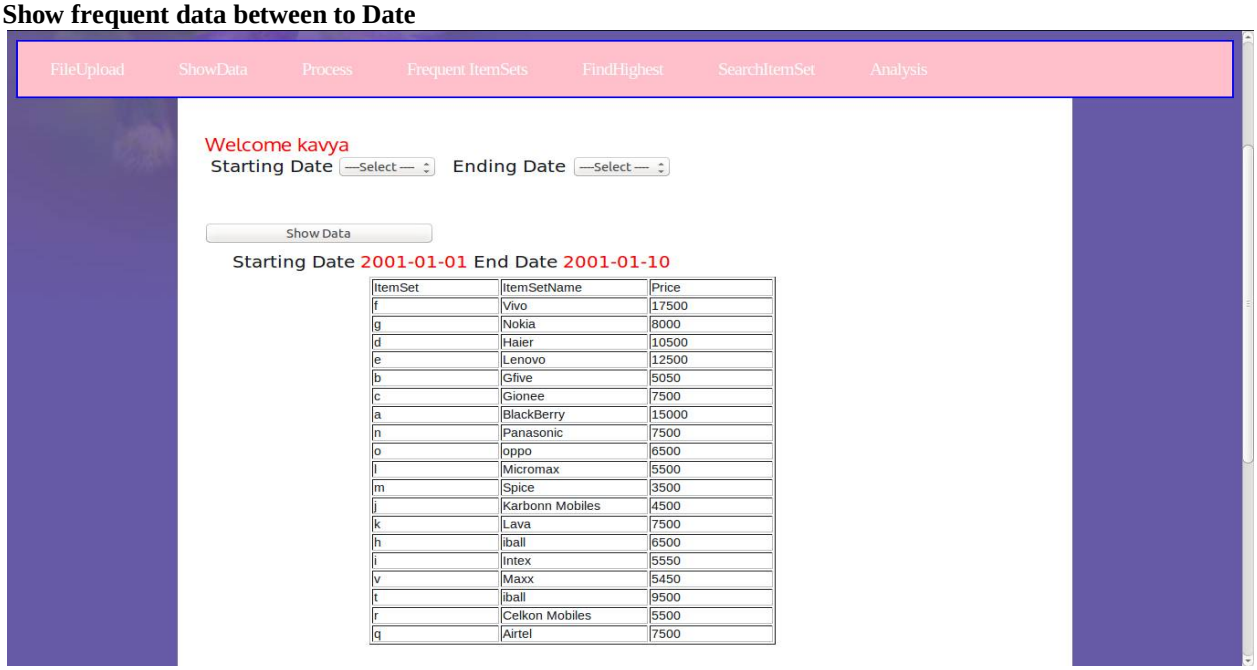


Figure 2 : File Upload

Show frequent data between to Date



ItemSet	ItemSetName	Price
f	Vivo	17500
g	Nokia	8000
d	Haier	10500
e	Lenovo	12500
b	Gfive	5050
c	Gionee	7500
a	BlackBerry	15000
n	Panasonic	7500
o	oppo	6500
l	Micromax	5500
m	Spice	3500
j	Karbonn Mobiles	4500
k	Lava	7500
h	iball	6500
i	Intex	5550
v	Maxx	5450
t	iball	9500
r	Celkon Mobiles	5500
q	Airtel	7500

Figure 3: Frequent Itemsets

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Sql Injection Attack

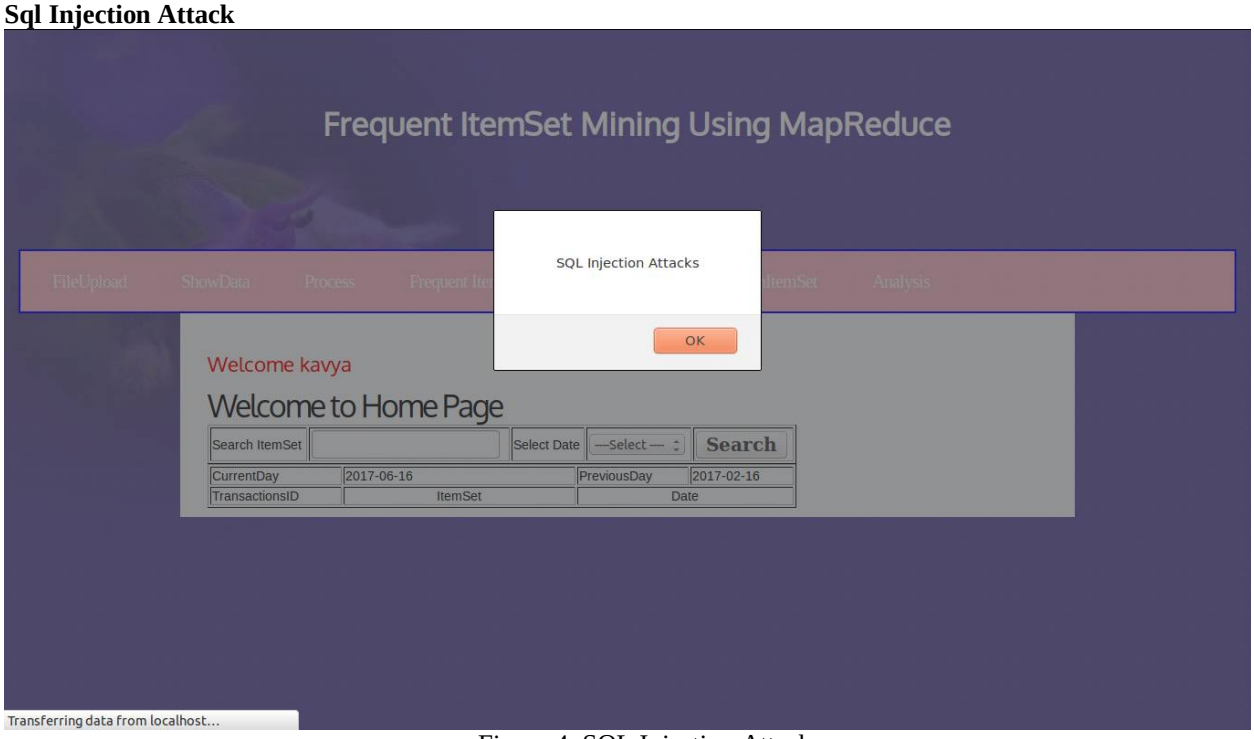


Figure 4: SQL Injection Attack

Figure 5: Time Show of MA and FIUT

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

V. EXPERIMENTAL RESULTS

The performance of this system against the Frequent Itemset Ultrametric Tree (FIUT)method is as shown in fig 5. Modified Apriori algorithm is a good algorithm to give the correct results as compared to existing systems. Also this algorithm shows the faster execution even for large database. As synthetic dataset is used here, so user can make their own dataset to run the tests. The implementation cost for this testing is negligible.

Size of Dataset	FIUT	FIMMA
1000	136	109
2000	161	149
3000	210	197
5000	270	210
10000	311	280

Table 1 : Time comparison

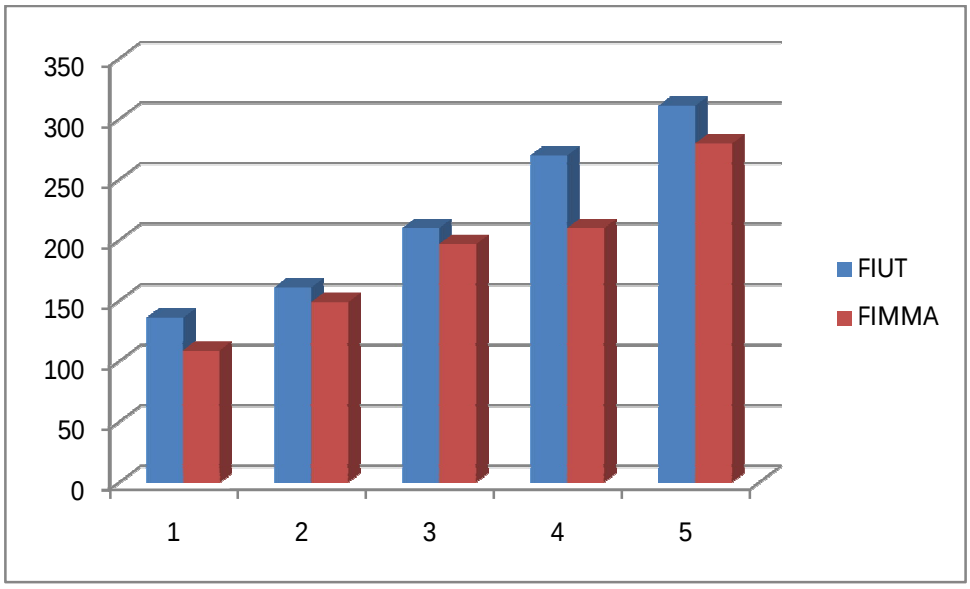


Figure 6: Size of Dataset.

As shown in table 1, first column shows the Size of Dataset. Other columns show the methodologies to find frequent itemset. As per this table, it shows the time required to extract frequent itemsets. Proposed system requires less time compare to existing systems. Support is 10 percent of aggregate records. It shows the time in seconds.

VI. CONCLUSION

Frequent Itemset Mining has attracted plenty of attention but much less attention has been given to mining Infrequent Itemsets. The mining of frequent itemset is an important research area in the field of data mining. The association rules are formed using the frequent itemset mined. Many different methods have been proposed for mining the frequent itemsets. The paper proposes a Mapreduce based Modified Apriori to find frequent itemsets using the Hadoop framework. Modified Apriori algorithm utilizes the procedure of hash tables in which it stores past values to compare

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

with new itemsets. It utilizes buckets in hash tables for storing results. Each buckets stores frequent items and will give a unique name, hence no need to scan repeatedly.

REFERENCES

[1]JW.Han, J.Pei and YW.Yin, ―Mining Frequent Patterns without Candidate Generation‖, International Conference on Management of Data, vol. 29(2), 2000, pp. 1-12.
[2] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Chang, ―PFP: Parallel FP-growth for query recommendation‖, Proceedings of the 2008 ACM Conference on Recommender Systems, 2008, pp. 107-114.
[3] Zhigang Zhang, GenlinJi, Mengmeng Tang, ―MREclat: an Algorithm for Parallel Mining Frequent Itemsets‖, 2013 International Conference on Advanced Cloud and Big Data.
[4] Hui Chen, Tsau Young Lin, Zhibing Zhang and JieZhong, "Parallel Mining Frequent Patterns over Big Transactional Data in Extended MapReduce‖, 2013 IEEE International Conference on Granular Computing.
[5] Xinhao Zhou, Yongfeng Huang, "An Improved Parallel Association Rules Algorithm Based on MapReduce Framework for Big Data‖, 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery.
[6]Jinggui Liao, Yuelong Zhao, Saiqin Long, ―MRPrePost-A parallel algorithm adapted for mining big data‖, 2014 IEEE Workshop on Electronics, Computer and Applications.
[7] SheelaGole, Bharat Tidke, ― Frequent Itemset Mining for Big Data in social media using ClustBigFIM algorithm‖, International Conference on Pervasive Computing.
[8] Siddique Ibrahim S P, Priyanka R, ―A Survey on Infrequent Weighted Itemset Mining Approaches‖ , 2015, IJARCET, Vol.4, pp. 199-203.
[9] SurendarNatarajan, SountharrajanSehar, ―Distributed FP-ARMH Algorithm in Hadoop Map Reduce Framework‖, 2013 IEEE.
[10] Xiaoting Wei, Yunlong Ma , Feng Zhang, Min Liu, WeimingShen, Incremental FP-Growth Mining Strategy for Dynamic Threshold Value and Database Based on MapReduce‖, Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design.