Efficient High Average-Utility Pattern Mining For Big Data

R. Vasumathi, Dr. S. Murugan

Abstract: High utility item set mining is an important topic in recent research of data mining. Frequent pattern mining is determined using average-utility mining with the utilities and profit. There have been lots of researches conducted on High Average-Utility Item set Mining. But most of the researches the data are stored in a centralized database. So in this work we propose a method High utility Item set Mining for Big data. The major issues in analyzing large data are execution time and memory space. So, we try to implement a new algorithm Efficient High Average-Utility Pattern Mining for Big Data (EHAUPMBD) using Hadoop platform to minimize the execution time and memory space. We have conducted experiments on real-time server datasets (RSD) and compare various parameters with existing algorithms.

Index Terms: Data Mining, High Utility Item set Mining, Big Data, Hadoop, and Map Reduce

1. INTRODUCTION

A process in which the interesting patterns and the knowledge are extracted from the large dataset is called Data Mining [1].Frequent item set mining is another important research topic in data mining. Frequent item set mining uses only frequent item sets and it does not taken into account the purchase quantities and unit profits. Hence, the informations extracted using frequent item set mining is not adequate for different applications. To extract meaningful information high utility item set mining was developed [2].High utility item set mining considers both profit and quantity. In high utility item set mining [HUIM] length of each item set is not taken and to overcome this problem High Average- Utility item set mining [HAUIM] was developed [4].Shifeng Ren proposed an efficient algorithm for High Average Utility. This author also proposed two tighter upper-bounds to minimize the search space [6].Hence, High average utility item set mining is not used to mine large scale data sets. We propose a new algorithm for big data. Hadoop is the open-source software that process vast amount of data. It provides a software framework for distributed storage and the big data is processed by the Map-Reduce programming model. In this paper, the real-time dataset with different the data size is analyzed by the Hadoop tool. The data file size has been used to analyze the performance with the various algorithms where the proposed algorithm performs well in accuracy, precision, recall, F-Measure, and processing time.

2. BACKGROUND OF THE STUDY

In data mining, association rule and frequent item set mining are important operations [3]. In many applications, the item set with the occurrence of the high frequency in the database has been studied [10]. Vincent et al.Proposed a new algorithm (EFficient High Utility Item set Mining) EFIM which depends on two upper-bound Viz sub-tree and local utility. These algorithms effectively reduce the Search Space [7].

- *R. Vasumathi, Dr. S. Murugan*
- *Full Time Research Scholar, PG & Research Department of Computer Science, Nehru Memorial College (Autonomous), Puthanampatti, Trichirapalli-621007, Tamilnadu, India*
- *Associate Professor, PG & Research Department of Computer Science, Nehru Memorial College (Autonomous), Puthanampatti, Trichirapalli-621007, Tamil Nadu, India.*
- *Email: rsvasumathi.msc@gmail.com*

J.C.W.Lin et.al. Proposed an algorithm HAUI-MMAU which uses two efficient methods to reduce the search space. The first method is Improved Estimated Utility Co-occurrence Pruning Strategy (IEUCP) using different join operations. Second method Pruning Before Calculation Strategy (PBCS) is developed to prune the item sets without scanning the data base [5]. [6]The author proposed an algorithm using two tighter upper-bound as an alternate to auub method. It largely reduced the Search Space. Jerry Chun-Wei Lin et.al. use multiple minimum high average utility thresholds to reduce the search space [8]. Tseng et.al. Proposed a new framework for Mining High Utility Item set Mining in big data. A new algorithm is PHUI-Growth is proposed for parallel mining HUIs on Hadoop platform. PHUI-Growth has high performance on large scale data sets [9]. In [10] the author proposed an algorithm to find High Average-Utility Item sets using Average-Utility (AU) list structure. This determined High Average-Utility Item sets efficiently and without candidate generation. The search space is minimized using downward closure property and increases the speed of discovering patterns.

3. EHAUPMBD Algorithm

In this algorithm, the total utility is calculated from the input data base table. Then the minimum average utility threshold value is calculated. The High Average Utility upper Bound Items are analyzed where the high average utility item sets are found. If the average utility is greater than or equal to total utility is multiplied by the threshold value then the Map-Reduce algorithm is applied. If the map stage is found, the key-value pairs separated from a set of data. Next, the reduce stage is integrated into the same set of key-value pairs. Now the transactions are reduced using the k-value.

Algorithm 1 EHAUPMBD Algorithm

Input: D is a transaction database table, δ is a user-specified threshold, MAU is a minimum average-utility threshold value, TU is Total Utility and AU is Average Utility.
Output: reduced set of transactions.

1 calculate the TU in D.
2 calculate the MAU Threshold.
3 Find High average utility itemset in D.
4 if HAUUBI is less than TU * δ then step 1
5 apply map-reduce
6 map-stage:

All the Key-value pairs separated from a set of data
7 Reduce Stage:
Integrated all the same set of Key-value pairs
8 Transactions reduced in D.

Pseudo code 1.EHAUPMBD algorithm

The proposed algorithm EHAUPMBD is compared with the existing algorithms, EFIM, HAUI_MMAU, MEMU, and MHUI_BD. The parameters used in this proposed system are accuracy, Precession, Recall, F-Measure, Processing Time (MS) and power consumption (MW).

4. EXPERIMENTAL RESULT

This paper analyzes the Hadoop tool with the data size starting from 100KB to 1000KB. In Data Mining, accuracy is one of the vital parameters. It refers to the number of corrected prediction over the total number of prediction. The high accuracy indicates the high stability of the data mining procedures. The table1 shows the calculated percentage of accuracy. The proposed algorithm (EHAUPMBD) achieved 91.69% of accuracy. Other methods like EFIM, HAUI-MMAU, MEMU, and MHUI_BD achieved 76.94%, 79.74%, 85.42%, and 87.85% respectively. The comparison graph of these methods is shown in figure 1.

Table.1 Percentage of Accuracy

RECS	EFIM	HAUI_ MMAU	MEMU	MHUI _BD	EHAUPMBD
100K	76.97	79.63	86.12	87.54	90.75
200K	77.01	79.49	85.53	88.39	90.47
300K	77.53	79.35	85.79	86.69	90.06
400K	76.21	79.44	85.95	86.53	90.46
500K	77.11	79.69	85.25	87.12	91.09
600K	77.17	79.62	86.33	87.11	89.96
700K	77.25	80.1	85.93	87.97	91.54
800K	76.86	79.57	85.13	88.63	90.38
900K	76.9	79.74	86	88.07	91.54
1000K	76.94	79.74	85.42	87.85	91.69

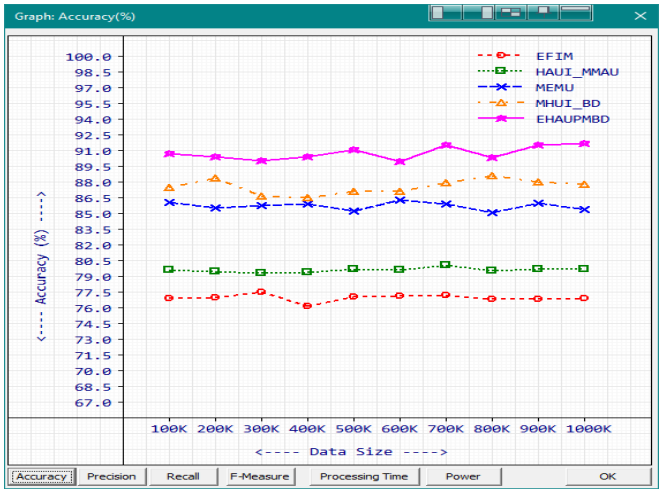

Figure .1 Graph Representation of accuracy rate

Table.2 Percentage of Precession

Recs	EFIM	HAUI_ MMAU	MEMU	MHUI _BD	EHAUPMBD
100K	77.54	79.86	85.77	88.42	91.54
200K	78.36	79.62	85.14	88.32	90.57
300K	78.04	79.31	85.33	86.29	89.55
400K	76.47	79.58	86.58	86.24	89.67
500K	77.42	79.24	85.23	86.08	90.41
600K	77.72	79.82	86.22	86.96	90.21
700K	78.13	80.08	85.67	87.99	91.4
800K	76.89	79.11	85.23	88.56	90.58
900K	77.76	79.5	85.87	88.18	91.63
1000K	76.75	79.83	85.02	87.13	91.43

Precession is one of the vital parameters in data mining. The precession is calculated for various sizes of the dataset is shown in table 2.The proposed algorithm achieved a precession of 91.43%. Other methods like EFIM, HAUI-MMAU, MEMU, and MHUI_BD achieved 76.75%, 79.83%, 85.02%, and 87.13% respectively. The comparative graph of these methods is shown in figure 2.

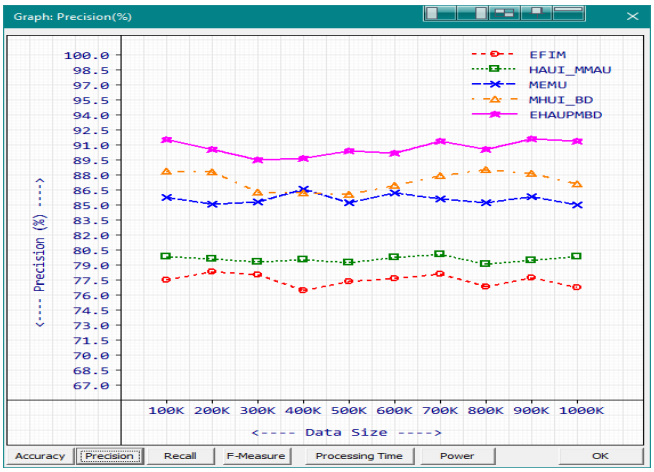

Fig. 2 Graph Representation of precession Percentage

Table .3 Percentage of recall

Recs	EFIM	HAUI_ MMAU	MEMU	MHUI _BD	EHAUPMBD
100K	76.66	79.49	86.37	86.88	90.11
200K	76.31	79.41	85.81	88.43	90.4
300K	77.25	79.37	86.12	86.99	90.47
400K	76.07	79.37	85.5	86.75	91.1
500K	76.94	79.96	85.26	87.91	91.66
600K	76.87	79.49	86.42	87.23	89.77
700K	76.77	80.11	86.11	87.95	91.67
800K	76.85	79.84	85.05	88.68	90.22
900K	76.44	79.88	86.1	88	91.47
1000K	77.04	79.69	85.71	88.39	91.91

Another significant parameter in data mining is recall. The percentage of recall is calculated and shown in table 3.The proposed algorithm achieved 91.91% of recall. Other methods like EFIM, HAUI-MMAU, MEMU, and MHUI_BD achieved 77.04%, 79.69%, 85.71%, and 88.39% respectively. The comparison of these methods is shown in figure 3.

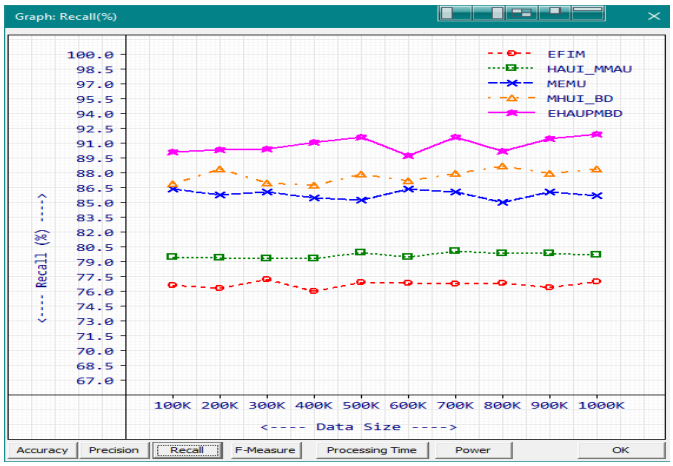

Figure .3 Graph Representation of recall percentage

Table. 4 Percentage of F-Measures

Recs	EFIM	HAUI_ MMAU	MEMU	MHUI _BD	EHAUPMBD
100K	77.0967	79.6728	86.0713	87.6443	90.818
200K	77.3201	79.5166	85.4733	88.3774	90.484
300K	77.644	79.3378	85.7243	86.6409	90.009
400K	76.2717	79.4727	86.038	86.4952	90.3795
500K	77.1807	79.5982	85.2428	86.9846	91.029
600K	77.2949	79.6567	86.3193	87.095	89.9895
700K	77.4446	80.096	85.889	87.9724	91.5327
800K	76.8708	79.4756	85.1406	88.622	90.3992
900K	77.0931	79.6873	85.9861	88.0875	91.5476
1000K	76.8961	79.7582	85.3614	87.7575	91.6683

Table.5 Processing Time in ms

Recs	EFIM	HAUI _MMAU	MEMU	MHUI_BD	EHAUPMBD
100K	15958	18238	18139	20129	15485
200K	31205	35953	36196	40003	30396
300K	46923	53743	54036	60037	45734
400K	62312	71836	72373	79555	60867
500K	78000	89869	90037	99766	76021
600K	93294	107619	108048	119387	91035
700K	108912	125395	126124	139182	106348
800K	124659	143299	144117	159170	121540
900K	140096	161329	162342	178896	136635
1000K	155512	178964	180026	199033	151694

F-Measure is also a significant parameter in data mining. F-Measure is calculated and shown in table 4.The proposed algorithm achieved 91.66% of F-Measures. Other methods like EFIM, HAUI-MMAU, MEMU, and MHUI_BD achieved 76.89%, 79.75%, 85.36%, and 87.75% respectively. The comparison graph of these methods is shown in figure 4.

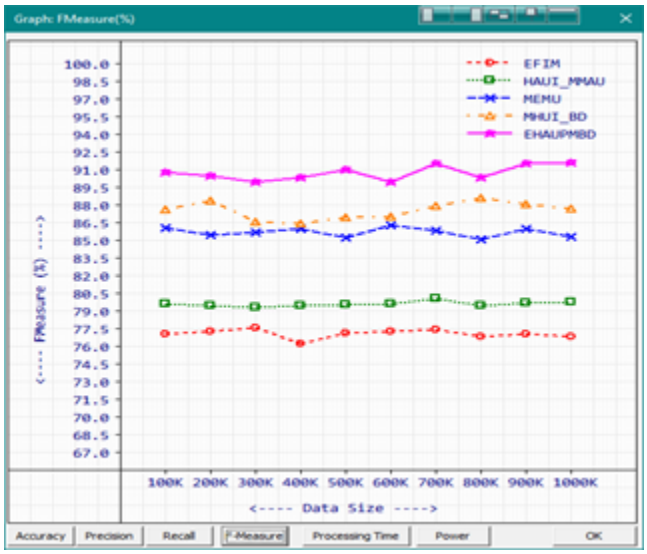

Figure. 4 Graph Representation of F-Measure

Processing time is the most important parameter in data mining. The processing time is calculated in milliseconds (ms). The processing time for proposed algorithm is 151694ms.The processing time for other methods like EFIM, HAUI-MMAU, MEMU, and MHUI_BD are 155512ms, 178964ms, 180026ms and 199033ms respectively. This is shown table 5. The comparison graph for processing time of these methods is shown in figure 5.

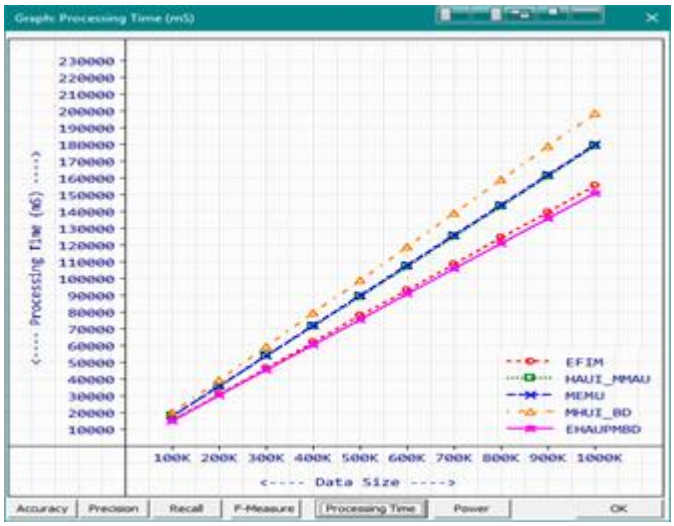

Figure. 5 Graph Representation of Processing Time

Table. 6 Power consumption in MW

Recs	EFIM	HAUI_ MMAU	MEMU	MHUI _BD	EHAUPMBD
100K	149	137	164	210	116
200K	246	301	334	439	235
300K	390	424	485	634	332
400K	508	546	644	841	461
500K	642	685	788	1077	572
600K	767	824	951	1278	681
700K	893	962	1103	1493	779
800K	1009	1080	1249	1696	884
900K	1132	1236	1409	1900	1005
1000K	1238	1359	1586	2123	1115

Power consumption is also calculated and shown in table 6.The power consumption for proposed algorithm is 1115MW.The power consumption for other methods like EFIM, HAUI-MMAU, MEMU, and MHUI_BD are 1238MW, 1359MW, 1586MW, and 2123MW respectively. The comparison graph of these methods is shown in figure 6.

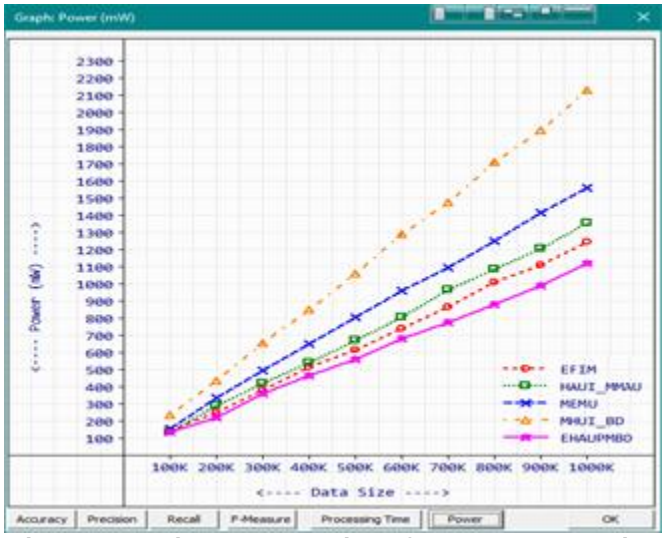

Figure.6 Graph Representation of power consumption

5. EXPERIMENTAL SETUP

To evaluate the performance of EHAUPMBD procedure RSD benchmark data sets are used, data size 3.59 GB, Total record size: 20971520. Accuracy, Precession, Recall, Processing time, and memory are measured with the RSD data sets. For comparative analysis existing methods EFIM, HAUI-MMAU,MEMU and HUIM-BD are taken. C++ programming language is used to implement the proposed algorithm.

6. CONCLUSION

This paper concludes that by analyzing the big dataset with the Hadoop tool by various data size in Kilobytes from 100 to 1000 gives the various results in the six parameters. This includes precision, recall, F-Measure, processing time and power consumption. In this analysis, the proposed algorithm gives the best performance in various parameters. The proposed algorithm with Hadoop tool performs well for big data (using RSD dataset).The proposed algorithm perform more reliable and flexible for analyzing big data sets with different file size.

REFERENCE

[1] Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. In ACM SIGMOD record 1993 (pp. 207-16). ACM.

[2] H.Yao,H.J.Hamilton, and C.J.Butz,“ A foundational approach To mining itemset utilities from databases,” in Proc. SIAM Int.Conf. Data Mining, 2004,pp.482-486.

[3] Krishnamoorthy S. Pruning strategies for mining high utility itemsets. Expert Systems with Applications. 2015; 42(5):2371-81.

[4] T.-P. Hong, C.-H. Lee, and S.-L. Wang, “Effective utility mining with the measure of average utility," Expert Syst. Appl., vol. 38, no. 7,pp. 8259_8265, 2011

[5] Lin W J.C, Li T, Fournier-Viger P, Hong TP, Su itemsets with multiple minimum thresholds. In the industrial conference on data mining 2016 (pp. 14-28). Springer, Cham.

[6] Jerry Chun-Wei Lin, Shifeng Ren, Philippe Fournier-Viger and Tzung-Pei Hong. ”EHAUPM: Efficient High Average-Utility Pattern Mining With Tighter Upper Bounds” Springer 5, 2017, PP 12927-12940.

[7] Souleymane Zida, Philippe Fournier-Viger, Jerry Chun-Wei Lin,Cheng-Wei Wu, Vincent. Tseng. ”EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining”Springer International Publishing Switzerland 2016. PP 14-28.

[8] Jerry Chun-Wei Lin, Shifeng Ren, and Philippe Fournier-Viger“MEMU:More Efficient Algorithm to Mine High Average- Utility Patterns With Thresholds” IEEE Volume 6, March 12, 2018.

[9] Ying Chun Lin, Cheng-Wei Wu, and Vincent S. Tseng. ”Mining High Utility Itemsets in Big Data” Springer International Publishing Switzerland 2015.PP 649-661.

[10] G.Grahne and J.Zhu, “Fast algorithms for frequent Itemset mining using FP-trees,” IEEE Trans.Knowl. Data Eng.,vol.17,no.10,pp.1347-1362, Oct.2005.