FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 614-619 | e-ISSN: 2347-470X

Auto-Threshold Dynamic Memory Efficient Frequent Pattern Growth for Data Excavation

G. Gunasekaran[1], S. Murugan[2] and K. Mani [2]

[1]Research scholar, PG & Research Department of Computer Science, Nehru memorial College (Autonomous) (Affiliated to Bharathidasan University), Puthanampatti-621007, Thiruchirappalli-Dt, Tamil Nadu, India Email: gunasekarangphd@gmail.com
[2,3]Associate Professor, PG & Research, Department of Computer Science, Nehru memorial College (Autonomous) (Affiliated to Bharathidasan University), Puthanampatti-621007, Thiruchirappalli-Dt, Tamil Nadu, India; [2]murugan_nmc@hotmail.com, [3]nitishmanil@gmail.com

***Correspondence:** G. Gunasekaran; Email: gunasekarangphd@gmail.com

ABSTRACT- Discovering patterns from large datasets is inevitable in the modern data driven civilization. Many research works, and business models are depending on this data excavation task. An efficient method for identifying and categorizing different data patterns from an exponentially growing database is required to perform a clear data excavation. A set of fresh processes such as Repeat Pattern Finder, Repeat Pattern Table, Repeat Pattern Threshold Analyzer, and Repeat Pattern Node are conceptualized in this work named as Auto-Threshold Dynamic Memory Efficient Frequent Pattern Growth for Data Excavation (AT-DME-FP). The main motive of this work is to improve the Accuracy, Precision, Recall, and F-Score along with the decrease in time and memory consumption. AT-DME-FP is contrived in a way to reduce the consumption of computational resources to match the modern data mining outgrowth. The memory reduction ability of AT-DME-FP makes it possible to use it with big data seamlessly.

Keywords: Auto-Threshold, Big data, Data Mining, FP-Growth, FP-Tree, Repeat Patterns, Repeat Pattern Table, Repeat Pattern Node.

ARTICLE INFORMATION
Author(s): G. Gunasekaran, S. Murugan and K. Mani

Special Issue Editor: Dr. S. Gopalakrishnan;
Received: 01/05/2022 **Accepted**: 03/06/2022; **Published**: 15/09/2022;
e-ISSN: 2347-470X;
Paper Id: 0322SI-IJEER-2022-15;
Citation: 10.37391/IJEER.100333
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100333.html

This article belongs to the Special Issue on **Intervention of Electrical, Electronics & Communication Engineering in Sustainable Development**

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Data excavation has vast usage in several applications such as Corporate Surveillance, Machine Learning, Market Analysis, and Customer behavior analysis [1]. Data excavation is used in cyber security and health care[2]. Now it is possible to predict upcoming health issues with a Machine Learning system. Data excavation plays a vital role in Customer Segmentation, Customer Relationship Management, Fraud detection, Financial Services, and Criminal Investigations. Hereditary based diseases are predicted using Data Mining procedures. The disease list starts from Diabetes to various types of cancers including Basal Cell Cancer, Breast Cancer, Prostate Cancer, Melanoma, Colon Cancer, Lung Cancer, Leukemia, and Lymphoma[3][4].The basic FP-Growth data mining procedure is carried over as the base of the proposed AT-DME-FP Procedure. [EMFIMD] [5], Rare Itemsets Mining Algorithm based on RP-Tree and Spark Framework [RIMA] [6], An Enhanced Accelerator Frequent Pattern Growth for Association Rule Mining [EAFPG] [7], and Mining Maximal Frequent Patterns in Transactional Databases and Dynamic Data Streams a Spark based approach [MFPTDBDDS] [8].

2. EXISTING METHODS

There are many formal data mining procedures are available such as Apriori, Eclat, FP-Tree, Naïve Bayes, and support vector machines. The patterns are discovered from the matrix and the Frequent Item Table (FIT) is created. Processing Time is analyzed in the results section but accuracy, precision, and recall are not analyzed in this work.

2.1 Efficiently Mining Frequent Itemsets on Massive Data [EMFIMD]

They are Efficiently Mining Frequent Itemsets on Massive Data. Precomputation-based Frequent Itemset Mining (PFIM) concept is used in the EMFIMD work. The PFIM creates the first large table with quasi-frequent itemsets that have higher supports than the lower bound. EMFIMD achieves a good processing speed but Accuracy, Precision, and Recall are not taken into consideration.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 614-619 | e-ISSN: 2347-470X

2.2 Rare Item sets Mining Algorithm based on RP-Tree and Spark Framework [RIMA]

RIMA is proposed based on RP-Tree and Spark Frame Work. Spark is an open-source domain-oriented web development framework developed using the programming language Java. RIMA starts the data mining process by arranging data vertically based on the transaction identifier. Then the vertical datasets are divided into two categories such as frequent vertical datasets and rare vertical datasets. Then RP-Tree algorithm is used to construct the frequent pattern tree. Processing time is measured for the datasets and discussed in the work, but other essential data mining parameters such as accuracy, precision, and recall are not discussed in this work.

2.3 An Enhanced Accelerator Frequent Pattern Growth for Association Rule Mining [EAFPG]

The EAFPG is designed It is claimed that the efficiency is increased by avoiding the conditional FP-Tree construction process which reduces the space and time during the frequent item sets detection from the database. The Accelerator Frequent Pattern Growth algorithm creates a binary matrix by scanning the database as the initiation.

2.4 Mining Maximal Frequent Patterns in Transactional Databases and Dynamic Data Streams: a Spark based approach [MFPTDBDDS]

MFPTDBDDS work targets business intelligence and Market Basket Analysis in particular. A prime number-based data transformation technique is used in MFPTDBDDS to reduce memory usage and faster execution. The transformed dataset is more relevant for the mining process after the annihilation of null values. MFPTDBDDS is designed to handle both static datasets and dynamic data streams.Accuracy, Precision, and Recall are not discussed in the results and analysis section. The computational overhead of MFPTDBDDS is $O(n) + O(m)$ where n is the items count and m is the prime assignments count.

3. PROPOSED METHOD

Auto-Threshold Dynamic Memory Efficient Frequent Pattern Growth for Data Excavation consists of three major functional modules. They are Repeat Pattern Finder (RPF), Repeat Pattern Table (RPT), Repeat Pattern Threshold Analyzer (RPTA), and Repeat Pattern Nodes (RPN). All these functional blocks are designed to reduce processing time and memory occupancy while retaining other essential characteristics such as accuracy, precision, and recall.

3.1 Repeat Pattern Finder

Repeat Pattern Finder module is used to find pattern similarities in an FP-Tree.

Table 1: Transaction Table

TID	Items	TID2	Items3
1	{a,b}	11	{c,j,k}
2	{b,c,d}	12	{c,k,l,m}
3	{a,c,d,e}	13	{c,j,l,m,n}
4	{a,d,e}	14	{c,j,m,n}
5	{a,b,c}	15	{c,j,k,l}
6	{a,b,c,d}	16	{c,j,k,l,m}
7	{a}	17	{c,j}
8	{a,b,c}	18	{c,j,k,l}
9	{a,b,d}	19	{c,j,k,m}
10	{b,c,e}	20	{c,k,l,n}

Latest studies regarding buyer pattern behavior show that majority of the buying patterns have oodles matches and correlations among them[11]. This lineament of purchase activity is creating redundant patterns with fiddling adaptation in a database[12]. Identifying the redundant patterns and handling them differently will reduce memory usage and improve the processing time. RPF is designed to search and find repeated patterns in an FP-Tree. A typical Frequent Pattern Tree is illustrated in *figure 1* for the transactions given in *table 1*. RPF uses pattern relations in match sets. A bottom-up approach is also used to identify partial matches. While observing the tree in *figure 1*, a repeated pattern can be easily identified as given in *figure 2*. The first structure starts with a $\emptyset$ (NULL) node whereas the second structure starts with an item C. While substituting the values $\emptyset \to C, a \to j, b \to k, c \to l, d \to m$, and $e \to n$, structure 2 can be called up using structure 1.

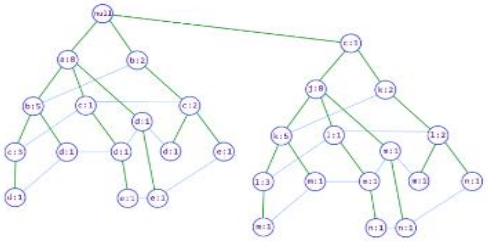


Figure 1: FP-Tree for Table 2 transactions

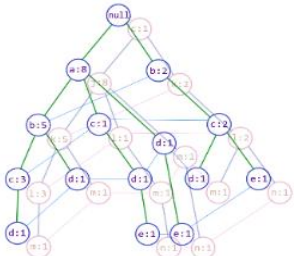


Figure 2: Structure matches of S1 and S2

The criterion definitions of RPF are as follows Let$S = \{s_0, s_1 \dots s_n\}$ where S is the set of different structures s_0, s_1 and so on. A match between two structures is declared when there

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 614-619 | e-ISSN: 2347-470X

is an equal number of nodes with similar interconnections. The root structure s_0 is defined as follows

$$s_0 \Rightarrow (\emptyset:\emptyset),(a:8),(b:2),(b:5),(c:1),(d:1),(c:2),(c:3),(d:1),(d:1),(d:1),(e:1),(d:1),(e:1),(e:1)$$

The structure S_1 is defined as

$$s_1 \Rightarrow (c:1),(j:8),(k:2),(k:5),(l:1),(m:1),(l:2),(l:3),(m:1),(m:1),(m:1),(n:1),(m:1),(n:1),(n:1)$$

3.2 Repeat Pattern Table

RPT is used to record the repeated patterns along with item variations lively during the pattern finding process. It consists of four fixed length fields and a variable length field. Pattern ID, Root Node ID, Number of nodes in a Repeated Structure (RS), and Number of Substitute Items Count are fixed with 16 bits, 64 bits, 8 bits, and 4 bits respectively. The Number of nodes in the RS field can hold up to 255 refers to a pattern that can have 255 nodes maximum including the root node. The fifth variable length field is based on the fourth field Substitute Item Count.

Table 2: Repeat Pattern Table entry for *figure 2*

Pattern ID	Root Node ID	Number of Nodes in RS	Substitute Item Count	Substitute Items
1	Address (null)	15 (00001111)	5 (0101)	j,k,l,m,n

If the value of the Substitute Item count is η the size of the fifth field will be $\eta \times 416$ times 416 bits. Since the size of Substitute Item Count is limited to 4 bits, the value of η goes up to 16. This refers to the number of substitute items that can be 16 at maximum. An example RPT is constructed for the structure matches of *figure 2* given in *table 2*.

3.3 Repeat Pattern Threshold Analyzer

RPTA maintains a separate temporary table concerning the pattern IDs. Each pattern in an FP-Tree is labelled with unique IDs and RPTA is used to find the memory allocation – memory saving ratio of introducing a repeat pattern node. Constructing a Repeat pattern node and replacing it is the FP-Tree consumes some processing time and memory. The purpose of RPTA is to filter less profitable RPT entries and RPN creations. Let T_{s_x} be the x^{th} repeat pattern memory threshold, then it is calculated as $T_{s_x} = \frac{\frac{M_a}{Mr}}{(Ma+M_r)} \times 100$ where M_a is the actual memory size of the entire pattern and M_r is the expected reduced memory value of the structure. Here M_r includes the 416 bits of RPT entry. The Acceptance function of RPTA operates based on the following equation.

$$Acceptance(T_{s_x}) = \begin{cases} accepted\ if\ \geq 0.15 \\ discarded\ otherwise \end{cases}$$

The value of 0.15 is finalized as cost cutting threshold of the RPTA after running an experiment with different datasets and justified with the minimum processing time.

3.3 Repeat Pattern Node

RPN is constructed for repeated pattern $s_x$once RPTA authorizes the T_{s_x} value. The RPN gets an entry in the AT-DME-FP table which represents the nodes in the Tree. The first bit of the AT-DME-FP Table is allocated to determine the node type, in which 0 refers to Ordinary Tree Node and 1 refers to Repeat Pattern Node. Other fields are used to refer to the Parent Address, Transaction Item-ID, and Count, Child node Count, and Child Node Addresses. AT-DME-FP table architecture is given in *table 3*.

Table 3: AT-DME-FP Table

Purpose	Size(bits)
Node Type	1
Parent Address	64
Transaction Item-ID	416
Count	64
Child Node Count (nc)	32
Child Node Address	nc * 64

After construction of repeat pattern Node PN1, the transaction tree represented will be reduced as in *figure 3*.

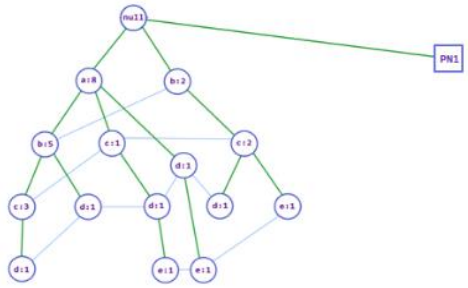


Figure 3: AT-DME-FP tree after construction of RPN

4. EXPERIMENTAL SETUP

To evaluate the performance of DME-FP procedure, benchmark datasets Chess, Kosarak[10], Mushroom[11][12], pumbsb_star[13], Retail[14] and T10I4D100K[15] are used. The processing time and memory consumption are measured in a computer with Processor 2.4 GHz i5-4210U Quad-core processor and 4GB RAM.

5. RESULTS AND ANALYSIS

Accuracy, Precision, Recall, Processing Time, and Memory [12] are measured with noted datasets. A comparative analysis of existing methods EMFIMD, RIMA, EAFPG, MMFPTD, and proposed AT-DME-FP is performed to measure the data mining parameters.

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 614-619 | e-ISSN: 2347-470X

5.1 Accuracy

Accuracy refers to the number of correct predictions over the total number of predictions.Higher accuracy indicates the higher stability of the data mining procedure. Measured values of accuracy for different methods are given in *table 4*.

Table 4: Comparison of Accuracy (%)

Accuracy (%)					
Data Chunk	**EMFIMD**	**RIMA**	**EAFPG**	**MMFPTD**	**AT-DME-FP**
1	84.18	88.22	85.25	86.28	90.31
2	84.7	88.96	85.55	86.82	91.08
3	84.34	89.86	85.38	87.9	90.43
4	85.86	88.56	87.94	87.64	91.02
5	86.83	90.98	87.46	88.61	92.09
6	85.26	88.85	85.44	88.35	91.93
7	86.37	90.38	87.39	88.4	92.41
8	85.57	88.32	87.08	86.83	92.59
9	86.02	90.72	85.43	87.14	91.84
10	84.88	88.68	86.16	86.96	91.44
Average	85.40	89.35	86.30	87.49	91.51

While calculating the average accuracy values for 10 different data chunks, AT-DME-FP gets 91.51% of accuracy, followed by 89.35% of RIMA, 87.49% of MMFPTD, 86.30% of EAFPG, and 85.40% of EMFIMD.Higher accuracy values reflect the rare itemset mining ability of a data mining procedure. Based on the observations, the proposed AT-DME-FP has the highest accuracy value of 92.59% while processing the 8th data chunk. The least accuracy value occurred while processing the 1st data chunk. The least accuracy score of AT-DME-FP is higher than the highest scores of other methods taken into comparison. The observed accuracy values are plotted as a graph for easy visual comparison – given in *figure 4*.

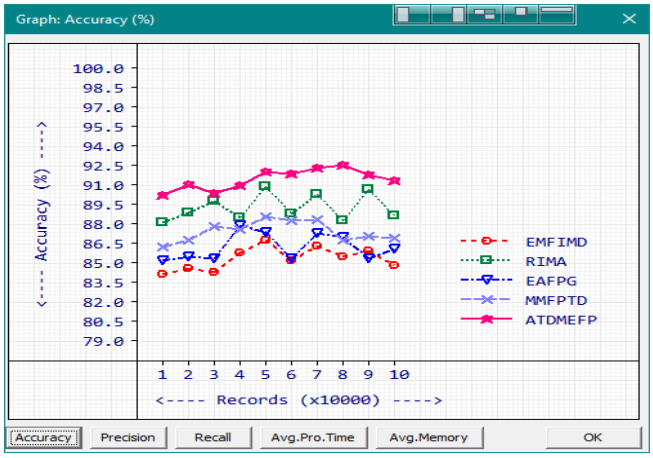


Figure 4: Accuracy (%)

5.2 Precision and Recall

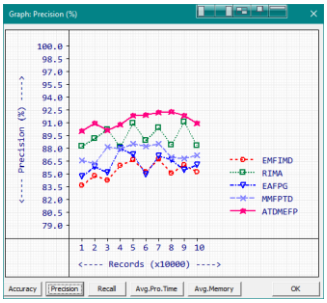
The precision score determines the reliability of data mining procedure. Higher precision refers to higher reliability. The calculated precision values are given in *table 5*.

Table 5: Comparison of Precision (%)

Precision (%)					
Data Chunk	**EMFIMD**	**RIMA**	**EAFPG**	**MMFPTD**	**AT-DME-FP**
1	83.78	88.3	84.82	86.65	90.17
2	84.86	89.24	85.96	86.35	91.06
3	84.34	90.31	85.27	88.23	90.2
4	86.05	88.25	88.13	88.01	90.89
5	86.78	91.06	87.35	88.63	91.91
6	85.33	89.01	85	88.32	91.98
7	86.84	90.54	87.24	88.61	92.31
8	85.15	88.46	86.78	87.08	92.4
9	86.17	91.19	85.55	86.91	91.94
10	85.32	88.41	86.19	87.28	91.06
Average	85.46	89.47	86.22	87.60	91.39

Recall refers to the fraction of successfully retrieved relevant items and it is an important parameter in data mining. A good mining procedure should score higher recall values. The recall is calculated using the formula

$$\frac{True\ Positive}{True\ positive + False\ Negative}$$



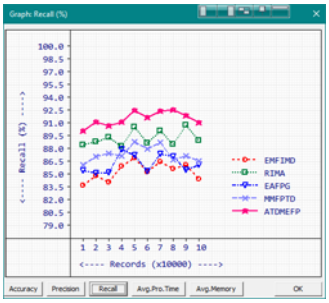


Figure 5(a): Precision (%) **Figure 5(b):** Recall (%)

The highest precision average of 91.39% is achieved by AT-DME-FP. The highest and lowest precision values of AT-DME-FP are 92.4% and 90.17% in order. RIMA scored the precision average value of 89.47% which comes closer to the proposed method. The remaining EMFIMD, EAFPG, and MMFPTD are getting a precision average of 85.46%, 86.22%, and 87.60% respectively. The prevision average values comparison graph is given in *figure 5(a).*

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 614-619 | e-ISSN: 2347-470X

Table 6: Comparison of Recall (%)

Recall (%)					
Data Chunk	**EMFIMD**	**RIMA**	**EAFPG**	**MMFPTD**	**AT-DME-FP**
1	83.79	88.48	85.47	86.14	90.13
2	84.86	88.85	85.17	87.17	91.15
3	84.13	89.36	85.27	87.49	90.73
4	86.03	88.3	87.93	87.2	91.16
5	87.01	90.62	87.25	88.87	92.5
6	85.36	88.71	85.38	88.04	91.7
7	86.51	90.15	87.47	88.79	92.43
8	85.71	88.59	87.17	86.73	92.62
9	86.13	90.83	85.54	87.24	91.94
10	84.52	89	86.16	86.64	91.12
Average	85.40	89.28	86.28	87.43	91.54

As per the calculations, the highest recall average is achieved by AT-DME-FP with a value of 91.54%. Other methods EMFIMD, RIMA, EAFPG, and MMFPTD are getting the recall average values of 85.40%, 89.28%, 86.28%, and 87.43% respectively. The highest recall value of AT-DME-FP is 92,62% and the lowest value is 90.13%. The Recall averages are plotted as a graph and given in *figure 5(b)*.

5.3 Processing Time

Processing time is an important factor that determines the quality of a data mining algorithm in this rapidly running big data world. The analytical results are expected on time and a good data mining procedure should consume a reasonable processing time. Measured processing times are given in *table 7*.

Table7:Comparison of Processing Time Average (mS)

Processing Time Average (mS)					
Data Chunk	**EMFIMD**	**RIMA**	**EAFPG**	**MMFPTD**	**AT-DME-FP**
1	3266	2853	3040	2528	2357
2	3360	2816	3204	2559	2257
3	3372	2982	3073	2675	2407
4	3244	2979	3101	2524	2381
5	3291	2910	3129	2647	2388
6	3329	2838	3067	2596	2227
7	3201	2868	3015	2674	2343
8	3400	3007	3122	2537	2386
9	3299	2930	3069	2600	2260
10	3306	2896	3206	2697	2409
Average	3306.8	2907.9	3102.6	2603.7	2341.5

Processing time is also considered a prime computational resource which is charged hourly basis in many of the server providers. The lowest processing time consumption average of 2341.5mS is achieved by AT-DME-FP. It is not exceeding more than 2409 mS in any situation while processing the 10 different data chunks. MMFPTD secured the second position with the processing time consumption average of 2603.7 mS.

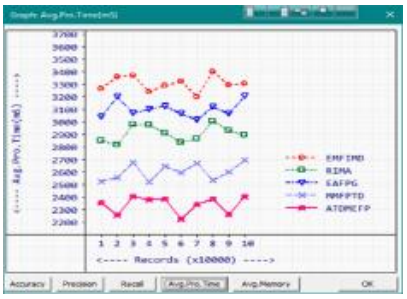


Figure 6: Processing Time averages (mS)

The remaining EMFIMD, RIMA, and EAFPG are getting processing time averages of 3306.8 mS, 2907.9 mS, and 3102.6 mS respectively. Processing time averages for processing 10 different data chunks by existing and proposed methods are plotted as a graph- shown in *figure 6*.

5.4 Memory

Memory is an important computational resource and the consumption of memory has a direct impact on the performance of any data mining algorithm. The lesser occupancy of the memory refers to the greater quality of the algorithm. The memory utilization characteristics are not limited to the data mining algorithms but to all computational procedures such as searching and sorting. Measured memory consumption values are tabulated in *table 8*.

Table 8: Comparison of Memory (B)

Memory (B)					
Data Chunk	**EMFIMD**	**RIMA**	**EAFPG**	**MMFPTD**	**AT-DME-FP**
1	1828	1443	1639	1354	1201
2	1828	1429	1631	1333	1217
3	1842	1435	1659	1333	1207
4	1862	1437	1643	1350	1207
5	1854	1457	1642	1345	1212
6	1831	1448	1633	1350	1249
7	1865	1427	1657	1319	1232
8	1864	1441	1618	1345	1236
9	1842	1419	1663	1339	1248
10	1843	1445	1665	1367	1201
Average	**1846**	**1438**	**1645**	**1343**	**1221**

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 614-619 | e-ISSN: 2347-470X

The threshold-controlled operation of AT-DME-FP is performed well in the case of memory consumption. It consumes 1249 Bytes of memory as the highest memory consumption which is the lower value of all other methods.

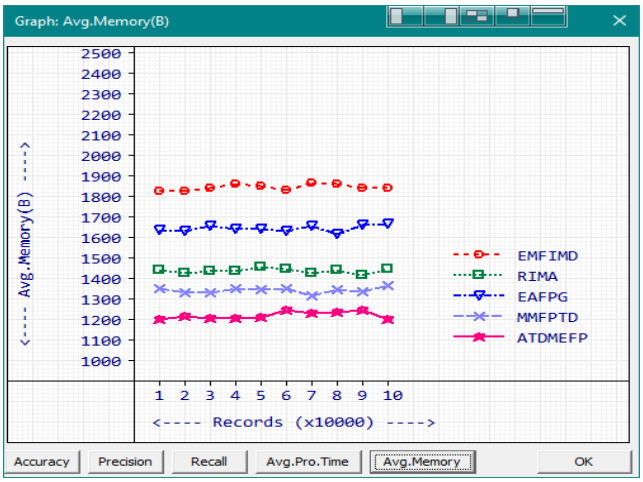


Figure 7: Memory consumption Average (B)

The memory consumption average of AT-DME-FP is 1221 Bytes, whereas EMFIMD, RIMA EAFPG, and MMFPTD are consuming 1846 Bytes, 1445 Bytes, and 1665 Bytes and 1343 Bytes respectively. The lower memory consumption average values show the efficiency of the AT-DME-FP procedure.

6. CONCLUSION

AT-DME-FP data mining procedure is made to handle modern big data seamlessly. Performing the data mining process with higher accuracy, precision, and recall makes it ineluctable to use AT-DME-FP. The Auto-Thresholding property makes the AT-DME-FP adaptable to use with several types of data. It is also running with limited computational resources such as processing time and memory. In general, improving the data mining accuracy will cost more computational resources whereas, attempting to minimize the resource consumption will reflect a reduction in accuracy, precision, and recall. But the proposed AT-DME-FP data mining procedure achieves higher accuracy, precision, and recall with reduced computational resources. Thus, AT-DME-FP can efficiently serve modern data mining requirements.

ACKNOWLEDGMENTS

The author would like to appreciate the effort of the editors and reviewers. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] Mendes, R., & Vilela, J. P. (2017). Privacy-preserving data mining: Methods, metrics and applications. In IEEE Access. IEEE Publications, 5, 10562–10582.
[2] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine learning and data mining methods in diabetes research. In Computational and Structural Biotechnology Journal. Elsevier, 15, 104–116.
[3] Dotson, J. P., Fan, R. R., Feit, E. M., Oldham, J. D., & Yeh, Y. (2017). Brand attitudes and search engine queries. In Journal of Interactive Marketing. Elsevier, 37, 105–116.
[4] Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., & Gao, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. Neurocomputing. Elsevier, 256, 56–62.
[5] Tarek, S., Abd Elwahab, R. A., & Shoman, M. (2017). Gene expression based cancer classification. In Egyptian Informatics Journal. Elsevier, 18(3), 151–159.
[6] Han, X., Liu, X., Chen, J., Lai, G., Gao, H., & Li, J. (2019). Efficiently mining frequent itemsets on massive data. In IEEE Access. IEEE Publications, 7, 31409–31421.
[7] Liu, S., & Pan, H. (2018). Rare itemsets mining algorithm based on RP-Tree and spark framework. In AIP Conference Proceedings. Australian Institute of Physics, 1967(1), 1–8.
[8] Rezaul Karim, Md., Cochez, M., & Beyan, O. D. (2018). Chowdhury Farhan Ahmed amd Stefan Decker, "Mining maximal frequent patterns in transactional databases and dynamic data streams: A spark-based approach" in Information SciencesVolume 432 (pp. 278–300). Elsevier.
[9] Resheff, Y. S., & Shahar, M. A statistical approach to inferring business locations based on purchase behavior. In IEEE International Conference on Big Data (Big Data), IEEE 2018 (pp. 2295–2303).
[10] http://fimi.ua.ac.be/data/kosarak.dat
[11] https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.names
[12] https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/
[13] http://fimi.ua.ac.be/data/pumsb_star.dat
[14] http://fimi.ua.ac.be/data/retail.dat
[15] http://fimi.ua.ac.be/data/T10I4D100K.dat
[16] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). 'Data Mining: Practical Machine Learning Tools and Techniques', Morgan Kaufmann publishers, MK (pp. 1–654).