Indonesian Journal of Electrical Engineering and Computer Science
Vol. 29, No. 3, March 2023, pp. 1355~1363
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v29.i3.pp1355-1363

Performance analysis of frequent pattern mining algorithm on different real-life dataset

Rakshit Khajuria[1], Anuj Sharma[1], Sunny Sharma[2], Ashok Sharma[2], Jyoti Narayan Baliya[3], Parveen Singh[4]
[1]Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India
[2]Department of Computer Science and IT, University of Jammu, Jammu, India
[3]Department of Educational Stduies, Cental University of Jammu, Jammu, India
[4]Department of Computer Science and IT, Cluster University of Jammu, Jammu, India

Article Info

Article history:

Received Jun 23, 2022
Revised Oct 26, 2022
Accepted Oct 29, 2022

Keywords:

Data mining
estDec+ algorithm
Frequent pattern mining
Pre-post+ algorithm

ABSTRACT

The efficient finding of common patterns: a group of items that appear frequently in a dataset is a critical task in data mining, especially in transaction datasets. The goal of this paper is to look into the efficiency of various algorithms for frequent pattern mining in terms of computing time and memory consumption, as well as the problem of how to apply the algorithms to different datasets. In this paper, the algorithms investigated for mining the frequent patterns are; Pre-post, Pre-post+, FIN, H-mine, R-Elim, and estDec+ algorithms. These algorithms have been implemented and tested on four real-life datasets that are: The retail dataset, the Accidents dataset, the Chess dataset, and the Mushrooms dataset. From the results, it has been observed that, for the Retail dataset, estDec+ algorithm is the fastest among all algorithms in terms of run time as well as consumes less memory for its execution. Pre-post+ algorithm performs better than all other algorithms in terms of run time and maximum memory for the Mushrooms dataset. Pre-Post outperforms other algorithms in terms of performance. And for Accident datasets, in terms of execution time and memory consumption, the FIN method outperforms other algorithms.

Corresponding Author:

Jyoti Narayan Baliya
Department of Educational Studies, Central University Jammu
Jammu, India
Email: jnbaliya2015@gmail.com

1. INTRODUCTION

The amount of data generated and collected from numerous sources has skyrocketed in recent years. Data mining is an interdisciplinary academic topic that has emerged to analyze massive amounts of data [1]. Many data mining applications rely heavily on frequent patterns. Thus, in order to obtain higher algorithmic efficiency and a better comprehension of the mined results, it is required to create algorithms for mining frequent patterns and learning the properties of the targeted data. How to find viable applications for the algorithm, is also required to examine the performance of various data mining methods on diverse datasets [2]. The focus of this literature survey is on the analysis of different frequent pattern mining algorithms on various real data sets. Many strategies for detecting common patterns have been proposed. Authors have taken different real-life data sets as well as to evaluate the performance of the algorithms against running time as well as memory consumption.

The N-list data representation, which is derived from the pre-order post-order code (PPC) tree, is an frequent patterns (FP) tree-like coding prefix tree that preserves critical information about frequent item-sets [3]. Algorithm performance is measured against the amount of time it takes to run and the amount of memory it consumes. Pre-post is the fastest in the majority of cases, according to the results. Pre-post+, a highperformance approach for mining frequent item-sets. It also includes a time-saving pruning approach known as children-parent equivalence pruning, which significantly reduces the search space [4]. On a variety of real datasets, rigorous tests were conducted to compare Pre-post+ against three state-of-the-art algorithms: Pre-post, FIN, and FP growth. Pre-ost+ is the fastest in the majority of cases, according to the results. Deng and Lv [5] proposed a node-set, a data structure for frequent mining item-sets that is more efficient. The paper compares FIN growth to Pre-post and FP growth on several real and synthetic datasets. The results show that FIN performs well in terms of both operating time and memory use. Developed a simple as well as novel data structure with the help of hyperlinks, H-struct, as well as mining algorithms. In this paper author do a study of H-mine against FP growth and Apriori algorithms and their performance is compared against running time and memory consumption [6]. The results reveal that H-mine performs admirably with diverse types of data. Recursive elimination for finding frequent item-sets. Relim is inspired by FP growth and the H-mine algorithm [7]. It does not need prefix trees or any other sophisticated data structures to process transactions [8]. Its main strength is its simplicity of its structure nit its speed. Paper presents an evaluation of recursive elimination over FP growth and éclat and Apriori on various datasets. Recursive elimination, which is based on deleting items, recursive processing, and reassigning transactions, performs well in tests. It's a quick and straightforward process to put in place.

Kaushal and Singh [9] using web click stream datasets, conducts a groundbreaking comparison study of five of the most essential sequential pattern mining techniques. Algorithm performance is measured in terms of execution time and memory consumption. Baralis *et al.* [10] has presented that it is economical and linearly scalable for extensive databases for both sparse and dense data distributions. Moreover, it outperforms FP growth in terms of performance. Deng and Wang [11] present a revolutionary vertical algorithm termed PPV for quick, frequent pattern discovery. This paper compares PPV with FP growth. The results of the experiments reveal that PPV is a good algorithm that outperforms FP growth, éclat, as well as dEclat. Deng *et al.* [12] described a new data representation called NC-set, which keeps track of the complete information used for erasable mining item-sets. Based on this NC set, a new algorithm has been proposed called MERIT for mining erasable item-sets efficiently. Deng [13] presented a new algorithm neural tangent kernel (NTK) to mine top-rank-k frequent patterns since mining Top-Rank-k regular patterns is an emerging topic in frequent pattern mining.

a) Data mining

While the data collected from diverse sources are often too large to be analyzed manually, computer technology has substantially increased in terms of processing power and storage capacity [14]. As a result, using computer technology to find information as well as knowledge in ever-increasing amounts of data has become feasible, inexpensive, and crucial. These factors necessitate the development of innovative approaches to convert large quantities of target data into meaningful information as well as knowledge in a sufficient length of time.

As a result, a new research topic is known as data mining, or database knowledge discovery has emerged. Data mining is the process of extracting nontrivial, previously unknown, as well as possibly useful information from enormous amounts of data [15]. It reflects the merging of numerous sciences, including machine learning, information theory, as well as database systems, as an interdisciplinary research subject. Association rule mining, classification, clustering, regression, as well as outlier detection are some of the most typical data mining activities [16].

b) Frequent pattern mining

Such patterns are pruned by frequent pattern mining tools, which consider them to be unwanted or of little interest. Because of its usefulness in so many domains, data mining has gotten a lot of interest in the database research community [17]. Fused deposition modeling (FDM) is a widely used data mining technique that is useful for various applications, including association mining, correlations, sequential item-sets, max-item-sets, partial periodicity, and emergent item-sets. FP are item-sets, subsequences, or substructures that emerge in the target dataset with a frequency more significant than a (user-defined) threshold value [18].

Naik [19] described mining for market basket analysis to solve the problem of mining association rules. The fundamental purpose of discovering association rules is to predict consumer behaviour by identifying inherent relationships between the many things that customers have purchased from retailers or supermarkets. Numerous works on the rapid mining of recurrent patterns can be divided into two groups: The first category is Apriori has presented by [20], and the second category is FP-growth and tree-projection has presented by [21]. In some situations, these tactics are still problematic has presented by [22]. Frequent pattern mining has been effectively implemented for enhanced decision support in a wide range of real-world applications, including shown in Figure 1.

With the globalization of trade, businesses encounter an increasing number of clients and transactions. As a result, they must be aware of both risks and possibilities. Mining common patterns can aid in the creation of promotions, discounts, retail layouts, exceptional marketing, storage management, and market forecasting has presented by [23]. In disaster prevention, analyzing numerous environmental parameters such as temperature, humidity, as well as wind, especially for imminent wind, can help forecast the weather as well as avoid loss as well as casualties has presented by [24], [25].

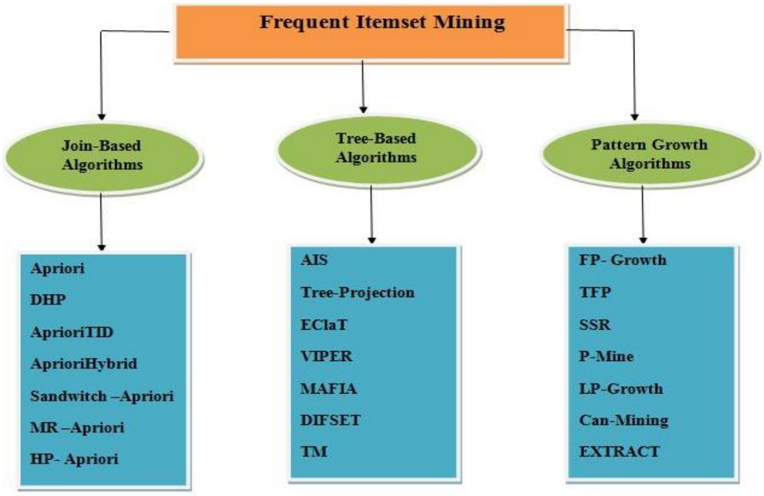

Figure 1. Classification of frequent pattern mining algorithms

2. ABOUT THE DATASET USED FOR EXPERIMENTATION

In this paper, we have used 4 public real-life datasets-Accidents, Chess, Mushrooms and Retail- to evaluate the mining algorithms, which includes the Pre-post algorithm, Pre-post+ algorithm, FIN algorithm, H-mine algorithm, Recursive elimination nestDec+ algorithm. Datasets has downloaded from the UCI repository and FIMI repository. The retail dataset comprises data from an anonymous retail store's market basket. It has 88,162 transactions incidents dataset that offers a wealth of information about different types of accidents and their causes. It contains 340,183 numbers of transactions. The mushrooms dataset offers information about several types of mushrooms. It includes 8,416 numbers of transactions. The chess dataset contains the different gaming steps having a probability of winning and losing. It contains 3,196 numbers of transactions. These are the real-life datasets taken to check the behaviour of the algorithms that which of these algorithms take less execution time and consume less memory for mining the most common patterns.

3. METHOD AND PROPOSED WORK

In this paper, we have conducted a comparative analysis of various recent frequent patterns mining algorithms by employing different datasets and then data mining is done using different pattern mining algorithms. The multiple datasets used in our study are Chess, Accidents, Mushrooms and Retail. The evaluation is done on the basis of run time as well as memory consumption and finally, the analysis has been done, and the results have been taken to conclude which among these selected algorithms performs better on what kind of dataset. Figure 2 presents the process of experimentation.

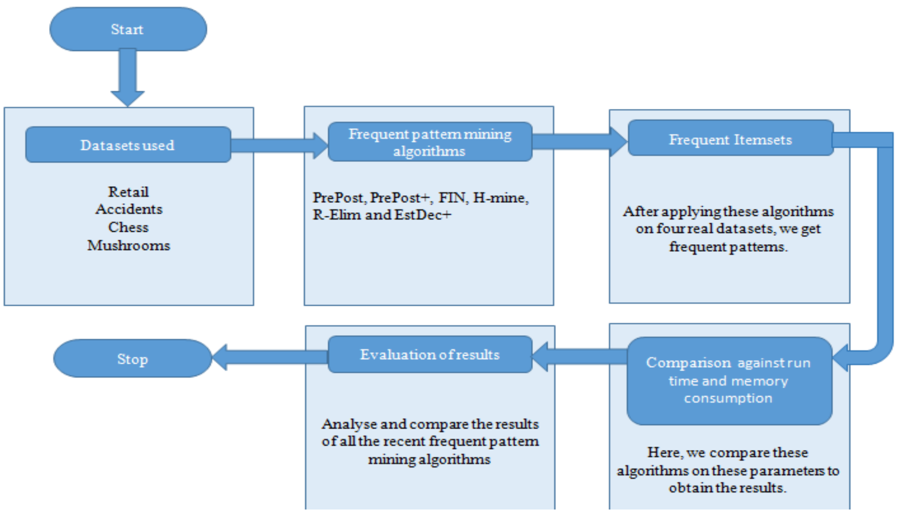

Figure 2. Process of experimentation

The majority of previously suggested algorithms for frequent mining item-sets can be divided into two groups: Apriori and FP growth. Despite the fact that several algorithms have been devised, one of the many significant research topics that have yet to be solved is how to build effective mining algorithms. Because the proposed approaches challenge the main memory requirement and efficiency of the occasions like dense vs. sparse, massive vs. memory-based data sets.

4. RESULTS AND DISCUSSIONS

The result shows the performance of every frequent pattern mining algorithm; the algorithm must deal with various real as well as synthetic data sets. In this paper, different experiments on real data sets are carried out to verify the algorithm's performance. Four real data sets have been utilised to evaluate the performance of many popular pattern mining methods: Pre-post algorithm, Pre-post+ algorithm, FIN algorithm, H-mine algorithm, Recursive elimination, and estDec+ algorithm. In Table 1 presents dataset parameters and its characteristics.

Table 1. Dataset parameters and characteristics

Dataset	Number of transactions	Distinct items	Size of a typical transaction	Real-world dataset
Retail	88162	16470	10.3	UCI repository
Accidents	340183	468	33.8	FIMI repository
Mushrooms	8416	119	23	UCI Repository
Chess	3196	75	37	UCI Repository

4.1. Running time or time complexity

The running time comparison of the algorithm on different data sets is shown in Figures 3-6. Figure 3 shows the running time of the compared algorithms on retail. Under all minimum supports, Pre-post outperforms the other six algorithms. Extensive minimum supports are required, estDec+ performs faster than Pre-post+ and different algorithms. However, estDec+ is faster than all algorithms, even when the minimum support is no more than 50%. At the support of 0.05, Pre-post runs fastest than all other algorithms. Figure 3 illustrates running time comparison for retail dataset.

Figure 5 displays the time it takes for the comparative algorithms to complete an accident. Pre-post is the most efficient algorithm, and it runs faster than Pre-post and FIN. The FIN algorithm fails to discover all frequent item-sets and runs out of memory when the minimum support is around 0.1 to 0.2. When the support is 0.1, Pre-post+ is still highly inefficient, taking over 10000s to complete. Pre-post+ outperforms both Pre-post and FIN algorithms with support of 0.3. FIN fails to find all frequent item-sets in a reasonable amount of time.

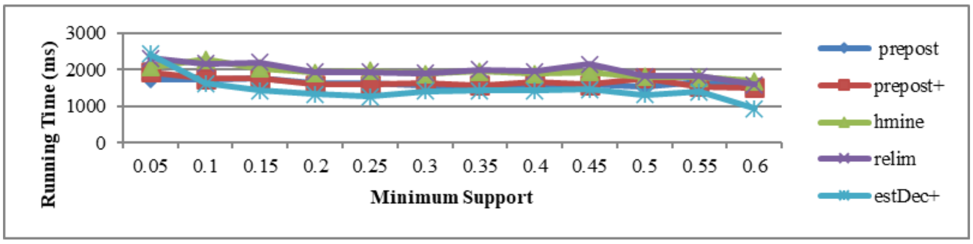

Figure 3. Running time comparison for retail dataset

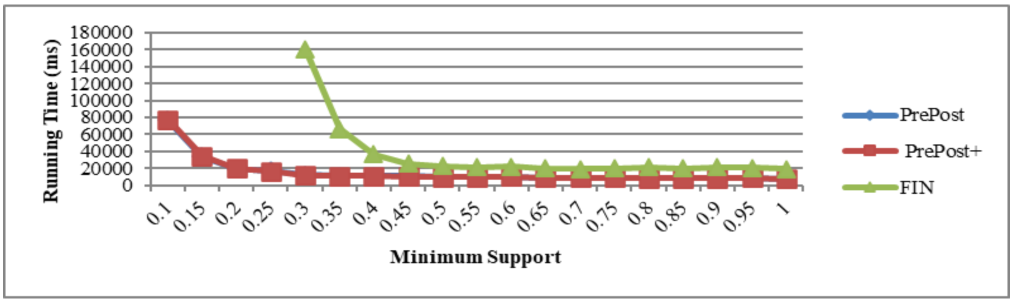

Figure 4. Running time comparison for accidents dataset

Figure 5 on Chess shows the running time of the comparative algorithms. At the support of 100%, Pre-post runs better overall the algorithms. At the support of 0.95, Pre-post runs the fastest and takes time 264s. At the support of 0.25 and onwards up to 0.35, Pre-post, FIN and H-Mine run out of memory and gives no result. At the minimum support of not more than 60%, Pre-post normally performs than others. H-mines take a lot of time to discover frequent item-sets.

Figure 6 on Mushrooms displays the running duration of the comparison algorithms. At the support of 100%, Pre-post and Pre-post+ perform much better than h-mine, R-Elim and FIN algorithms. At the support of 0.05 to 0.25, R-Elim runs out of memory and fails to adopt all frequent item-sets. At the support of 0.25, Relim takes a lot of time, like 392315s, which is a much longer time for finding the frequent item-sets. Pre-Post performs better at 50% of support than others but takes 957s for finding the frequent item-sets.

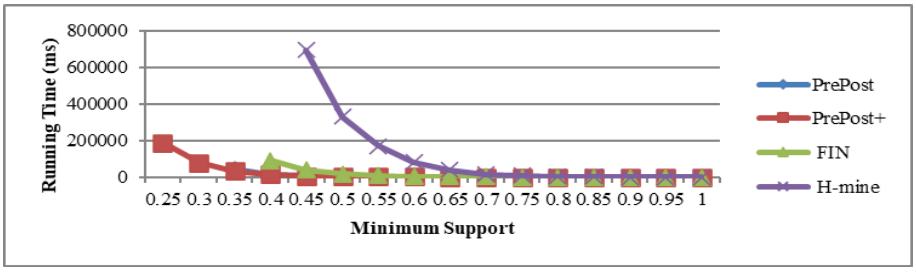

Figure 5. Running time comparison for chess dataset

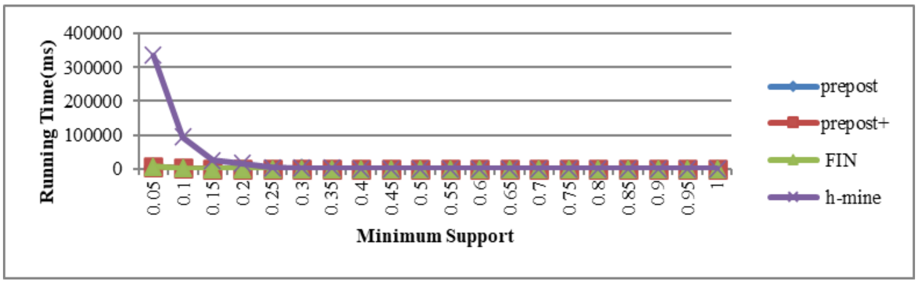

Figure 6. Running time comparison for mushrooms dataset

4.2. Maximum memory comparison result

Figure 7 on retail, which is a dense data set, reveals the memory cost of the contrasting techniques. The maximum memory consumption of the algorithm is compared on different data sets, as shown in Figures 8-11, respectively. EstDec+ consumes four to five times as much memory as H-mine and outperforms all other algorithms H-mine uses around 1.2 times the amount of memory that R-Elim does on average. Pre-post, as well as Pre-post+, consumes a large amount of memory. Nevertheless, the performance of Pre-post and Pre-post+ is almost the same. R-Elim performs even better than these two algorithms but consumes more memory than H-mine.

Figure 8 shows for the Chess dataset illustrates the memory use of the compared methods. H-mine consumes less memory than other methods when the support is 100%. As the level of assistance drops, Pre-post+ becomes worst, having a memory consumption of 780.4503 mb. At the support count from 0.05 to 0.3, Pre-post, FIN and H-mine give an out of memory error. As the minimum support increased from 0.05 to 1, the maximum memory value decreased good giving performance.

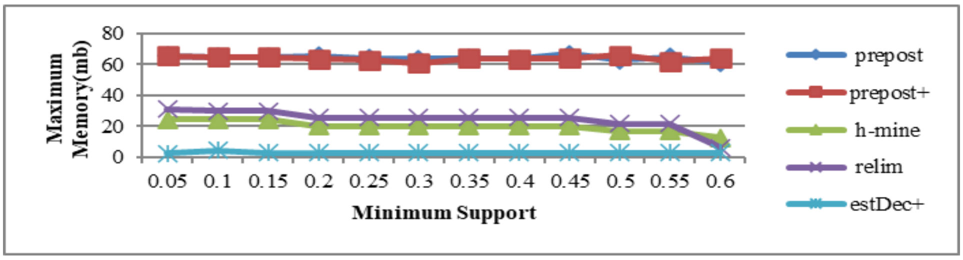

Figure 7. Memory usage comparison for retail dataset

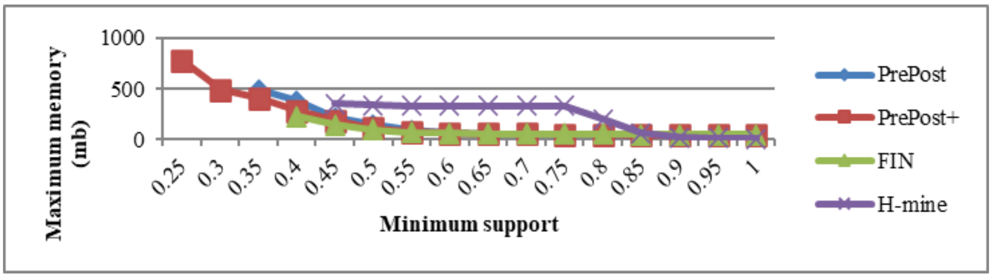

Figure 8. Memory usage comparison for chess dataset

Figure 9 displays a comparison of the memory usage of the compared algorithms for accident datasets. At the support of 100%, Pre-post+ performs much better than Pre-post, while FIN runs out of memory and fails to adopt the frequent patterns. But as we increase the support from 0.1 to so on, the maximum memory usage decreases, which is not good. When the support is not more than 60%, the Pre-post performs better than the other two algorithms. Figure 10 which is using the Mushrooms datasets illustrates the memory cost of the different techniques. The memory consumed by H-mine is two to three times faster than that of Pre-post, Pre-post+ and FIN algorithms and consumes about 5.3519 MB at the minimum support of 100%.

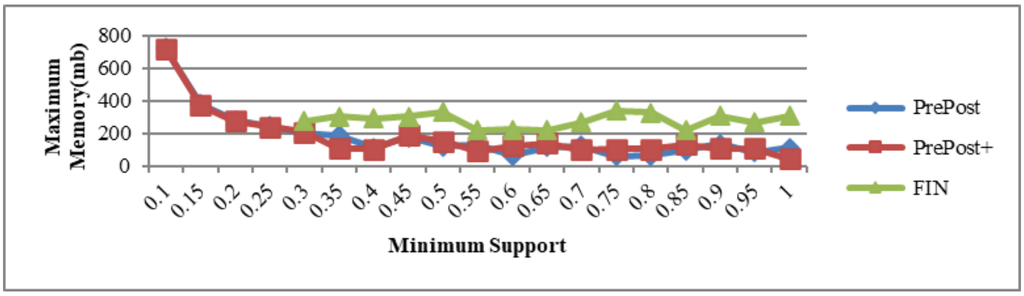

Figure 9. Memory usage comparison for accidents dataset

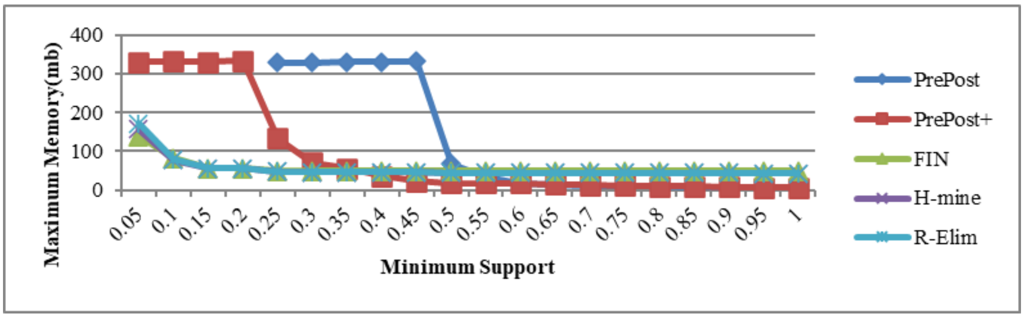

Figure 10. Memory usage comparison for mushrooms dataset

The Recursive elimination consumes slightly the same memory as that of the H-mine, which is about 5.88574 MB. As the support decreased from 100% to 10%, for example, the memory usage of all algorithms increased. H-mine, on the other hand, uses less memory than other algorithms. Recursive elimination runs out of memory at the support ranges from 0.05 to 0.2.

Considering the results from Figures 3-10, it can be concluded that out of the six algorithms, five algorithms have provided better results on two data sets which are retail and mushrooms. For accidents, Pre-post consumes the least time and Pre-post+ consumes the least amount of memory at 100% support. Likewise, Pre-post consumes the least time for chess and H-mine consumes less memory. In the same way, for mushrooms, Pre-post and Pre-post+ take the same time to produce frequent patterns. In Table 2 shows results of the compared algorithm over real-life dataset. In Table 3 shows comparision result of different algorithms over real-life datasets.

Table 2. Results of the compared algorithms over real-life datasets

Algorithms	Datasets	Run Time (ms)	Maximum Memory (MB)
Pre-post	Mushrooms	1510.85	53.74
	Chess	4547.5	89.23
	Accidents	15341.68	171.30
	Retail	979.05	38.63
Pre-post+	Mushrooms	1442.6	52.93
	Chess	17077.9	140.45
	Accidents	15380.63	171.77
	Retail	999.35	38.33
FIN	Mushrooms	1608.1	55.70
	Chess	8675.15	52.18
	Accidents	27309.10	211.11
	Retail	--	--
H-mine	Mushrooms	24838.75	88.89
	Chess	67977.5	133.69
	Accidents	--	--
	Retail	1162.1	11.96

Table 3. Results of the compared algorithms over real-life datasets

Algorithm	Datasets	Run Time (ms)	Maximum memory (mb)
R-Elim	Mushrooms	36434.06	92.0545
	Chess	--	--
	Accidents	--	--
	Retail	1191.75	14.68
estDec+	Mushrooms	--	--
	Chess	--	--
	Accidents	--	--
	Retail	874.25	1.75

The Tables 2-3 compared all the algorithms in terms of memory consumption and run time. In terms of execution time, the estDec+ algorithm is the fastest among all algorithms for each minimal support for the retail dataset. Pre-post+ algorithm is the fastest among all algorithms for the mushrooms dataset, whereas for the chess dataset, Pre-post performs well. Finally, for accidents datasets, FIN performs much better than Pre-post and Pre-post+. Also, in terms of the maximum memory, the estDec+ method is the quickest of all algorithms for each minimal support for the retail dataset. Pre-post+ algorithm is the fastest among all algorithms for mushrooms datasets whereas, for chess, FIN is better than all other algorithms. Finally, for accidents datasets, Pre-post and Pre-post+ perform the same as they have a difference of 0.30 MB.

5. CONCLUSION AND FUTURE WORK

Pre-post, Pre-post+, FIN, H-mine, and Recursive- Elimination, estDec+, are among the most useful frequent pattern mining algorithms in this study. The evaluation of these algorithms has been done on four real-life datasets. The algorithms' running time and memory consumption are compared, and the results are provided. It has been observed that for the Accidents dataset, Pre-post and Pre-post+ consume the least time. It has a novel N-list data structure that comes from an FP-tree-like coding prefix tree called PPC-tree that keeps the critical information about frequent item-sets and uses the least memory. Because transactions with the same prefixes share the same nodes of a PPC-tree, the N-list is compact. The difference between the two algorithms is the smallest. Pre-post out performs because the counting of item-sets is changed into the intersection of N-lists, reducing the complexity of the intersecting N-lists to O(m+n) by an efficient method. The dataset utilized determines the storage cost for maintaining the N-list of item-sets. Because the dataset employed here is dense, the storage cost is low.

Extending these algorithms to produce efficient ways for mining common item-sets is an exciting future direction for our research. Since the amount of data available is increasing at an exponential rate, using these algorithms to extract common item-sets from big data is also an intriguing task. We plan to use these techniques to find the most common item sets in terms of future extensions of this work. We'll aim to include all of the algorithms' ideas into the process of extracting patterns from large amounts of data. Parallel/distributed implementations of these algorithms are also an exciting task as the available data is growing exponentially. Moreover, the work can be done on different datasets for the sake of running time and memory consumption. It is necessary to offer a novel data structure for extracting all frequent patterns from transactional databases with a single database scan and without having to rescan the original database.

REFERENCES

[1] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” *SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, May 2000, doi: 10.1145/335191.335372.

[2] C. C. Aggarwal, M. A. Bhuiyan, and M. Al Hasan, “Frequent Pattern mining algorithms: a survey,” in *Frequent Pattern Mining*, C. C. Aggarwal and J. Han, Eds. Cham: Springer International Publishing, 2014, pp. 19–64, doi: 10.1007/978-3-319-07821-2_2.

[3] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, “Data mining techniques and applications – A decade review from 2000 to 2011,” *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11303–11311, 2012, doi: 10.1016/j.eswa.2012.02.063.

[4] Z.-H. Deng and S.-L. Lv, “Pre-post+: An efficient N-lists-based algorithm for mining frequent item-sets via Children–Parent Equivalence pruning,” *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5424–5432, 2015, doi: 10.1016/j.eswa.2015.03.004.

[5] Z.-H. Deng and S.-L. Lv, “Fast mining frequent item-sets using Nodesets,” *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4505–4512, 2014, doi: 10.1016/j.eswa.2014.01.025.

[6] J. Pei, J. Han, and W. Wang, “Constraint-based sequential pattern mining: the pattern-growth methods,” *J. Intell. Inf. Syst.*, vol. 28, no. 2, pp. 133–160, 2007, doi: 10.1007/s10844-006-0006-z.

[7] C. Borgelt, “Keeping things simple: Finding frequent item sets by recursive elimination,” in *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, 2005, pp. 66–70, doi: 10.1145/1133905.1133914.

[8] R. Davashi, “ILUNA: Single-pass incremental method for uncertain frequent pattern mining without false positives,” *Inf. Sci. (Ny).*, vol. 564, pp. 1–26, 2021, doi: 10.1016/j.ins.2021.02.067.

[9] C. Kaushal and H. Singh, “Comparative study of recent sequential pattern mining algorithms on web clickstream data,” in *2015 IEEE Power, Communication and Information Technology Conference (PCITC)*, 2015, pp. 652–656, doi: 10.1109/PCITC.2015.7438078.

[10] E. Baralis, T. Cerquitelli, and S. Chiusano, “Index support for frequent itemset mining in a relational DBMS,” in *21st International Conference on Data Engineering (ICDE’05)*, 2005, pp. 754–765, doi: 10.1109/ICDE.2005.80.

[11] Z. Deng and Z. Wang, “A new fast vertical method for mining frequent patterns,” *Int. J. Comput. Intell. Syst.*, vol. 3, no. 6, pp. 733–744, 2010, doi: 10.1080/18756891.2010.9727736.

[12] Z. Deng, Z. Wang, and J. Jiang, “A new algorithm for fast mining frequent item-sets using N-lists,” *Sci. China Inf. Sci.*, vol. 55, no. 9, pp. 2008–2030, 2012, doi: 10.1007/s11432-012-4638-z.

[13] Z.-H. Deng, “Fast mining top-rank-k frequent patterns by using node-lists,” *Expert Syst. Appl.*, vol. 41, no. 4, Part 2, pp. 1763–1768, 2014, doi: 10.1016/j.eswa.2013.08.075.

[14] P. K. Aggarwal, P. Jain, J. Mehta, R. Garg, K. Makar, and P. Chaudhary, “Machine learning, data mining, and big data analytics for 5G-enabled IoT,” in *Blockchain for 5G-Enabled IoT: The new wave for Industrial Automation*, S. Tanwar, Ed. Cham: Springer International Publishing, 2021, pp. 351–375, doi: 10.1007/978-3-030-67490-8_14.

[15] P. P. Jadhav, “11 - Advanced data mining for defense and security applications,” in *Artificial Intelligence in Data Mining*, D. Binu and B. R. Rajakumar, Eds. Academic Press, 2021, pp. 223–241, doi: 10.1016/B978-0-12-820601-0.00009-4.

[16] P. Bachhal, S. Ahuja, and S. Gargrish, “Educational data mining: A review,” *J. Phys. Conf. Ser.*, vol. 1950, no. 1, p. 12022, Aug. 2021, doi: 10.1088/1742-6596/1950/1/012022.

[17] A. A. Abdelaal, S. Abed, M. Al-Shayeji, and M. Allaho, “Customized frequent patterns mining algorithms for enhanced Top-Rank-K frequent pattern mining,” *Expert Syst. Appl.*, vol. 169, p. 114530, 2021, doi: 10.1016/j.eswa.2020.114530.

[18] M. T. Alam, C. F. Ahmed, M. Samiullah, and C. K. Leung, “Mining frequent patterns from hypergraph databases,” in *Advances in Knowledge Discovery and Data Mining*, 2021, pp. 3–15.

[19] S. B. Naik, “Mining associations rules between attribute value clusters,” in *Advances in Artificial Intelligence and Data Engineering*, 2021, pp. 909–917.

[20] M. B. Nichol *et al.*, “Quality of anticoagulation monitoring in nonvalvular atrial fibrillation patients: Comparison of anticoagulation clinic versus usual care,” *Ann. Pharmacother.*, vol. 42, no. 1, pp. 62–70, 2008, doi: 10.1345/aph.1K157.

[21] J. Pei *et al.*, “PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth,” *Proc. - Int. Conf. Data Eng.*, pp. 215–224, 2001, doi: 10.1109/icde.2001.914830.

[22] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, “H-mine: hyper-structure mining of frequent patterns in large databases,” in *Proceedings 2001 IEEE International Conference on Data Mining*, 2001, pp. 441–448, doi: 10.1109/ICDM.2001.989550.

[23] L. Cao, Y. Zhao, and C. Zhang, “Mining impact-targeted activity patterns in imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1053–1066, 2008, doi: 10.1109/TKDE.2007.190635.

[24] Z. Zhang, W. Wu, and Y. Huang, “Mining dynamic interdimension association rules for local-scale weather prediction,” in *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, vol. 2, 2004, pp. 146–149, doi: 10.1109/CMPSAC.2004.1342698.

[25] R. Khajuria, A. Sharma, A. Sharma, and P. Singh, "A survey on various approaches to examine cognitive behavior and academic performance of learner in virtual learning," *International Conference on Innovative Computing and Communications*, Springer, Singapore, 2023.

BIOGRAPHIES OF AUTHORS

Rakshit Khajuria received BTech in Computer Science and Engineering and M.Tech degree in Computer Science and Engineering from Shri Mata Vaishno Devi University, Katra. He is currently Pursuing Ph.D at the department of in Computer Science and Engineering, Lovely Professional University, Jalandhar, India. He has published research articles in International and Scopus journals. His current research interest is Machine Learning, Artificial Intelligence, and Academic Performance. He can be contacted at email: khajuriarakshit75@gmail.com.

Dr. Anuj Sharma working as Assistant Professor in School of Computer Applications at Lovely Professional University, Phagwara (Punjab). PG in Master of Computer Applications from VTU, Belgaum Karnatka. Ph.D in Computer Applications from Career Point Universit, Kota Rajasthan. Published papers in international, Scopus and SCI journals. He can be contacted at email: anuj.20592@lpu.co.in.

Dr. Sunny Sharma working as Lecturer at University of Jammu. He hasd published many papers in international, Scopus and SCI journals. He can be contacted at email: sunny202658@gmail.com.

Dr. Ashok Sharma is having 18 years of Teaching Experiences in Higher Education and he has worked in Various Reputed Institution of Higher Learning in India in different capacity. His area of interest is machine learning, cloud computing and data science. He is certified in Beidou Technology from Shanghai Jio Tong University, Shanghai, Leveraging Technology for Effective Teaching in the Classroom and Beyond from International Institute of Information Technology, Bangalore, Internet of Things. He can be contacted at email: drashoksharma@hotmail.co.in.

Dr. Jyoti Narayan Baliya working as Associate Professor and Head in Department of Educational Studies, Central University of Jammu. He is having Doctorate in Education along with Post graduate in three subjects i.e. Education, Mathematics and Sociology and many other academic credentials during his period of service. He has 32 years of Experience of research and teaching in Higher Education. He can be contacted at email: jnbaliya2015@gmail.com.

Dr. Parveen Singh working as Associate Professor, head of the Department, Department of Computer Sciences, Higher Education Department, Govt. of Jammu and Kashmir. He has Completed his Master and Ph.D from Department of Computer Science and IT, University of Jammu, Jammu. He can be contacted at email: iamparveen@yahoo.com.