4. Experimental Results and Discussion

The algorithms viz., Matrix-Apriori, VDF, NPF-VDF and TB-NPF-VDF were implemented using the Python programming language (version 3.8.2). To estimate the performance of TB-NPF-VDF, the research work used four real-time datasets downloaded from the FIMI repository and an open-source Data Mining Library. Table 21 describes the characteristics of datasets. The purpose of using these datasets is that they have been used as a reference by researchers primarily for FPM and ARM-based research. To do a uniform and fair comparison, the experiments for all the datasets of all algorithms were conducted using the same software and hardware configurations. The experiments were performed using 8.00GB RAM, Intel Core i7 with 2.40GHz 64-bit processor and Windows 8.1. All algorithms' runtime performance (Matrix-Apriori [7], VDF, NPF-VDF, TB-NPF-VDF) for the four datasets with different min_sup percentages ranging from 20% to 70% were tabulated in Table 22.

Datasets	Transaction count	Item count	Average item count/transaction
chess	3196	75	37.00
mushrooms	8416	119	23.00
T25i10d10k	9976	929	24.77
c20d10k	10000	192	20.00

Table 21. Characteristics of Datasets

min_sup (%)	Runtime (in Sec.)			
	Matrix -Apriori	VDF	NPF-VDF	TB-NPF-VDF
chess				
20	20.7578	16.8578	13.3578	6.5267
30	19.6365	16.0452	12.1455	5.0325
40	17.7750	14.0750	10.0720	4.5635
50	16.3028	13.3017	9.0017	3.2634
60	15.3625	12.7943	8.2934	2.4571
70	14.8546	11.9825	7.4822	2.0012
mushroom				
20	23.2135	21.1215	18.0016	12.1024
30	21.3426	20.0462	17.0642	11.5642
40	20.0035	19.7083	14.1038	10.7869
50	19.2002	18.2058	13.2044	10.0063
60	18.0805	17.7898	12.7240	8.5698
70	17.5652	15.9575	11.4530	7.9586
t25i10d10k				
20	25.2145	23.3254	20.3325	15.1267
30	23.9625	21.4578	19.4258	13.9568
40	21.5467	20.0025	17.9857	12.0127
50	20.3859	18.7621	16.2456	11.6321
60	19.5321	18.0056	15.0012	10.5212
70	18.4521	16.0527	13.7564	9.2451
c20d10k				
20	26.0014	24.4253	22.8342	17.7586
30	24.9532	22.6752	21.5062	15.9802
40	22.4251	21.9546	20.0412	13.7542
50	21.5621	19.4316	18.8562	11.9892
60	20.1425	19.0012	17.0124	11.0016
70	19.1478	17.5242	15.9351	10.0142

Table 22. Performance Results

Figures 2 to 5 show the graphical representation of the runtime comparison between the algorithms viz., Matrix-Apriori, VDF, NPF-VDF and TB-NPF-VDF for the datasets, namely chess, mushroom, t25i10d10k and c20d10k, respectively. From Table 22 and from figures 2 to 5, it was observed that the runtime performance of TB-NPF-VDF outperforms than Matrix-Apriori, VDF and NPF-VDF. On an average, the runtime performance is improved from 20.3092 to 9.9094.

Further, to prove statistically, a Welch two-sample *t*-test is being performed between the runtimes of Matrix-Apriori and TB-NPF-VDF. The test was done to determine whether the mean runtimes of Matrix-Apriori and TB-NPF-VDF are equal to each other or not. The null hypothesis is taken as that the two mean runtimes are equal, and the alternative is that they are not equal. The test is performed using the R tool for each dataset, and the results are tabulated in Table 23.

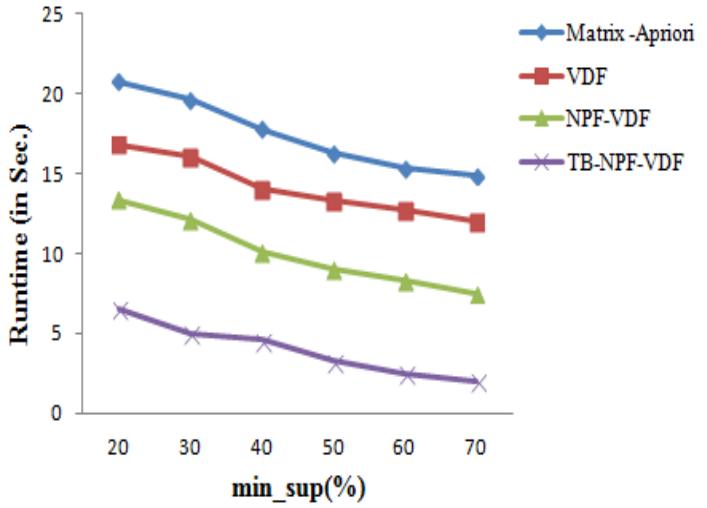

Fig. 2. The execution time of Matrix-Apriori, VDF, NPF-VDF and TB-NPF-VDF for chess dataset

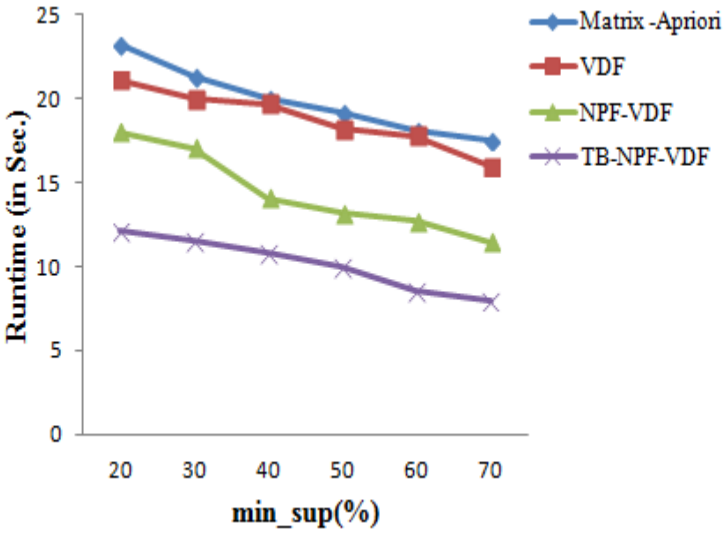

Fig. 3. The execution time of Matrix-Apriori, VDF, NPF-VDF and TB-NPF-VDF for mushroom dataset

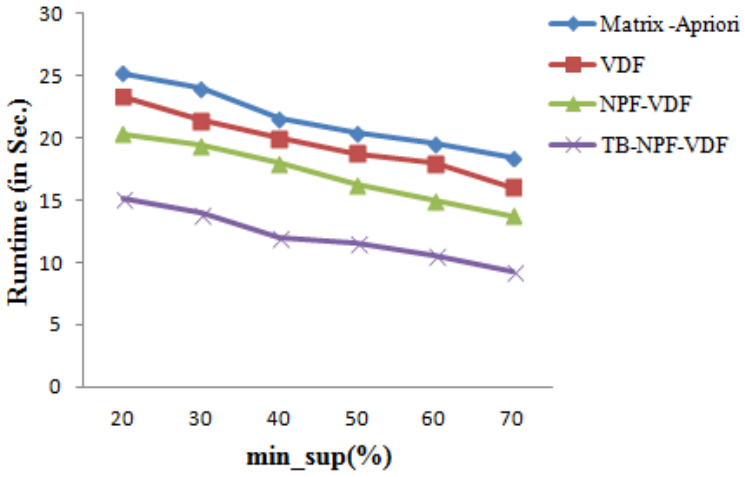

Fig. 4. The execution time of Matrix-Apriori, VDF, NPF-VDF and TB-NPF-VDF for t25i10d10k dataset

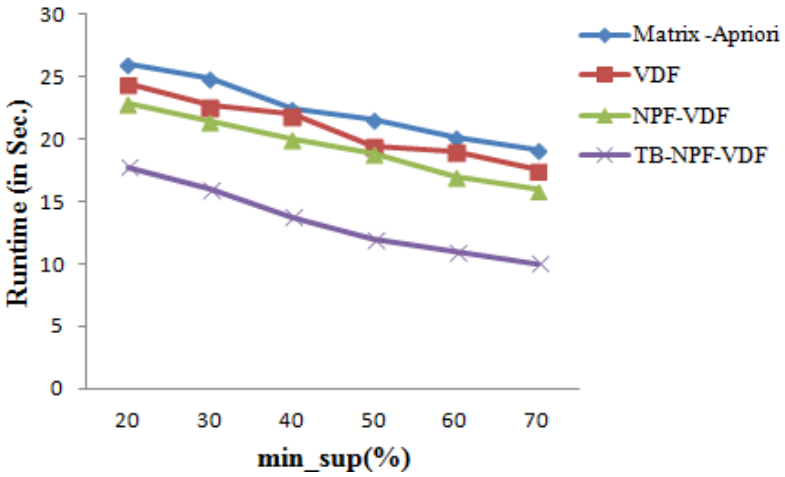

Fig. 5. The execution time of Matrix-Apriori, VDF, NPF-VDF and TB-NPF-VDF for c20d10k dataset

Dataset	p-value
chess	1.207×10^{-06}
mushroom	6.785×10^{-06}
t25i10d10k	5.611×10^{-05}
c20d10k	0.0002914

Table 23. Results of *t*-Test

From the observation of *t*-test results, it is noted that for all datasets, the p-value is ≤ 0.05 (5%) which concluded that the two means are not equal, which means that there are significant differences between the runtimes. Therefore, the proposed method TB-NPF-VDF is more efficient in terms of runtime than the others.

The reason for enhancing the performance is that the concurrent execution of the tasks using a multithreaded approach speeds applications up and reduced the time required for execution by utilizing the CPU effectively. With novel pattern generation, the set of candidate elements generated is less than the existing ones. Further, it scans the database only once during the entire process.

5. Conclusion

Many FPM algorithms were introduced in the field of data mining. Each algorithm has its own merits and demerits and is unsuited for all real-life situations. A new approach called TB-NPF-VDF has been introduced in this research article to discover the frequent patterns that efficiently combine the power of VDF, NPF, and multithread concepts. Experiments were carried out on real-time datasets using python implementation for the existing and proposed methods. TB-NPF-VDF has been proven to be superior to other sequential approaches through memory usage and run time. The main advantage is that it discovers frequent patterns with less time and saves memory with jagged array representation for the VDF matrix. In future, the work can be improved by applying new and efficient optimization techniques.

References

[1] Guo, Y. M.; Wang, Z. J. (2010): A vertical format algorithm for mining frequent item sets. Proceedings of 2nd International Conference on Advanced Computer Control (IEEE Xplore), **4**, pp. 11-13.
[2] Han, J.; Kamber, M.; Pei, J. (2011): *Data mining concepts and techniques*, 3rd edn. Morgan Kaufmann.
[3] Aqra, I.; Herawan, T.; Ghani, N. A.; Akhunzada, A.; Ali, A.; Razali, R. B.; Choo, K. K. R. (2018): A novel association rule mining approach using TID intermediate itemset. PloS one, **13**(1), pp. 01-32.
[4] Subhashini, A.; Karthikeyan, M. (2019): Itemset Mining using Horizontal and Vertical Data Format, International Journal for Research in Engineering Application & Management. **5**(3) pp. 534-539.
[5] Gawwad, M. A.; Ahmed, M. F.; Fayek, M. B. (2017): Frequent itemset mining for big data using greatest common divisor technique. Data Science Journal, **16**(25), pp. 1-10.
[6] Usha, D.; Rameshkumar, K. (2014): A Complete Survey on application of Frequent Pattern Mining and Association Rule Mining on Crime Pattern Mining. International Journal of Advances in Computer Science and Technology, **3**(4), pp. 264-275.
[7] Pavón, J.; Viana, S.; Gómez, S. (2006): Matrix Apriori: Speeding up the Search for Frequent Patterns. Databases and Applications, pp. 75-82.
[8] Sumathi, P.; Murugan, S. (2018): A Memory Efficient Implementation of Frequent Itemset Mining with Vertical Data Format Approach. International Journal of Computer Sciences and Engineering, **6**(11), pp. 152-157.
[9] Chon, K.W.; Hwang, S. H.; Kim, M. S. (2018): GMiner: A fast GPU-based frequent itemset mining method for large-scale data. Information Sciences, **439**, pp. 19-38.

[10] Huang, Y. S.; Yu, K. M.; Zhou, L. W.; Hsu, C. H.; Liu, S. H. (2013): Accelerating parallel frequent itemset mining on graphics processors with sorting. Proceedings of IFIP International Conference on Network and Parallel Computing, pp. 245-256.
[11] Huang, C. H.; Leu, Y. (2015): A LINQ-based conditional pattern collection algorithm for parallel frequent itemset mining on a multi-core computer. Proceedings of ASE BigData & Social Informatics, pp. 1-6.
[12] Zong-Yu, Z.; Ya-Ping, Z. (2012): A parallel algorithm of frequent itemsets mining based on bit matrix. Proceedings of IEEE International Conference on Industrial Control and Electronics Engineering, pp. 1210-1213.
[13] Tanna, P.; Ghodasara, Y. (2015): Analytical Study and Newer Approach towards Frequent Pattern Mining using Boolean Matrix. IOSR Journal of Computer Engineering, **17**(3), pp. 105-109.
[14] Jen, T. Y.; Marinica, C.; Ghariani, A. (2016): Mining frequent itemsets with vertical data layout in MapReduce. Proceedings of International Workshop on Information Search, pp. 66-82.
[15] Vijay Kumar, G.; Valli Kumari, V. (2013): Parallel Regular-Frequent Pattern Mining in Large Databases. International Journal of Scientific & Engineering Research, **4**(6).
[16] Gan, W.; Lin, J. C. W.; Fournier-Viger, P.; Chao, H. C.; Yu, P. S. (2019): A survey of parallel sequential pattern mining. ACM Transactions on Knowledge Discovery from Data (TKDD), **13**(3), pp. 1-34.
[17] Huynh, B.; Trinh, C.; Dang, V.; Vo, B. (2019): A parallel method for mining frequent patterns with multiple minimum support thresholds, International Journal of Innovative Computing. Information and Control, **15**(2), pp. 479-488.
[18] Qiu, H.; Gu, R.; Yuan, C.; Huang, Y. (2014): YAFIM: a parallel frequent itemset mining algorithm with spark. Proceedings of IEEE International Parallel & Distributed Processing Symposium Workshops, pp. 1664-1671.
[19] Shruti, I.; Abhay, K. (2018): Parallel Eclat with Large Data Base Parallel Algorithm and Improve its Effectiveness. International Journal of Engineering Trends and Technology, **60**(3), pp. 180-183.
[20] D. Kalpana, Data Mining Apriori Algorithm Implementation Using R, International Research journal of Engineering and Technology. **4**(11), pp. 1810- 1815.
[21] Sumathi, P.; Murugan, S. (2021): GNVDF: A GPU-accelerated Novel Algorithm for Finding Frequent Patterns Using Vertical Data Format Approach and Jagged Array. International Journal of Modern Education and Computer Science (IJMECS), **13**(4), pp. 28-41.

Authors Profile

P.Sumathi received her B.Sc and M.Sc degrees in Computer Science from Seethalakshmi Ramaswami College (affiliated to Bharathidasan University), Tiruchirappalli, India in 2001 and 2003 respectively. She received her M.Phil degree in Computer Science in 2008 from Bharathidasan University. She is presently working as an Assistant Professor in the Department of Computer Science, Vysya College, Salem. She is currently pursuing Ph.D, a degree in Computer Science in Bharathidasan University. Her research interests include Data Mining, Data structures and Database concepts.

S.Murugan received his M.Sc degree in Applied Mathematics from Anna University in 1984 and M.Phil degree in Computer Science from Regional Engineering College, Tiruchirappalli in 1994. He is an Associate Professor in the Department of Computer Science, Nehru Memorial College (Autonomous), affiliated to Bharathidasan University since 1986. He has 32 years of teaching experience in the field of Computer Science. He has completed his Ph.D degree in Computer Science with a specialization in Data Mining from Bharathiyar University in 2015. His research interest includes Data and Web Mining. He has published more than 25 research articles in reputed National and International journals.

V.Umadevi obtained her M.Sc degree in Computer Science & Information Technology and M.Phil degree in Computer Science from Madurai Kamaraj University. She has completed her Ph.D degree in Computer Science from CMJ University. Besides, she has received M.Tech and MBA degrees. She has 15 years of teaching experience in Computer Science. Her area of teaching and research interests include Management Information Systems, Project Management and Wireless Sensor Networks. She has published 28 research papers in National and International journals and authored three books. Also produced one Ph.D candidate. She has received National Award for "South Indian Achiever" in March 2020 and a "Lifetime Achiever" award from International Lions Club in March 2021. She has published a patent entitled "AI abetted material synthesising for hybrid metal rubber composite and 3D Printing" in August 2021.