International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No. 47, March 2025 – www.ijais.org

Efficient Mining of FP-Growth Algorithm Structure and Apriori Algorithm using OFIM for Big Data

Abdulkader M. Al-Badani
Faculty of Science and Engineering, Department of Computers, Aljazeera University, Ibb, Yemen

Abeer A. Shujaaddeen
Faculty of Computer Science & Information Systems, Sana'a University, Sana'a, Yemen

Motea M. Aljafare
Faculty of Computing and IT, University of Science and Technology, Sana'a, Yemen

ABSTRACT
Mining big data is difficult.Working with massive datasets requires the use of computer software and an efficient algorithm to solve problems.Data mining specialists are familiar with two popular association rule algorithms: apriori and fp-growth. Businesses are able to make well-informed decisions based on customer trends and behavior thanks to these algorithms' assistance in finding patterns and correlations within large datasets. These data mining procedures are now even more accurate and efficient thanks to developments in machine learning techniques. However, there are some disadvantages to the association rule approach, including the requirement for a lot of memory, the necessity for extensive dataset searches to ascertain the item set's frequency, and sometimes less-than-ideal rules. The authors of this study examined the fp-growth, Apriori, and OFIM algorithms in order to analyze the rule results of the three methods. Significant performance differences were found in the results, with the fp-growth algorithm showing the best efficiency when working with big datasets. On the other hand, the Apriori approach had scalability issues and frequently resulted in longer processing times as the amount of the dataset expanded, despite being simpler to construct.Changes to the FP_Growth algorithm's operation are proposed in this study.The suggested method would reduce mining time and the quantity of regularly created items by using the suggested matrix OFIM to construct a highly compact FP-tree, which would significantly lessen the amount of decision-making required for huge datasets.Furthermore, by minimizing the number of items generated often, our method optimizes memory usage and greatly increases its speed while processing huge datasets.

General Terms
Data Mining, Association Rule, Frequent Itemsets Mining.

Keywords
FP-Growth Algorithm, Aprioiri Algorithm, FP-tree, Support Count, OFIM

1. INTRODUCTION
The development of information technology in the modern era has made it necessary to develop methods that make the collection of extraordinarily large data sets easier, allowing for significant improvements in the amount of data that can be gathered and stored in a database [1]. In addition to improving the effectiveness of data collecting, these developments give businesses the ability to evaluate enormous volumes of data and derive insightful conclusions. Decision-making across a range of businesses now heavily relies on the ability to handle and interpret large amounts of data. The algorithm for association rules is one of the many branches. The association rule technique is used to mine frequently occurring itemsets and produce Boolean association rules [2]. Because it is the result of applying the characteristics of often occurring itemsets, it is known as an association rule. Since its introduction, the Apriori method has been refined and examined by numerous academics. The association analysis of the Apriori algorithm is, of course, now much more effective after enhancements. Alternative methods, including the FP-Growth approach, which greatly improves performance and eliminates the requirement for candidate generation, have been developed as a result of these developments. Consequently, scholars persist in investigating diverse approaches to enhance data mining procedures and extract more profound understandings from extensive datasets.

For association rules, there is also the well-known FP-Growth algorithm [3]. When compared to the initial association rule algorithm, the FP growth algorithm has considerably decreased the method's association analysis time. To overcome the problem of multiple scans, it uses a tree structure to generate itemsets rather than repeatedly searching the database[4]. The technique can mine data more efficiently thanks to this tree structure, called the FP-tree, which makes it possible to store and retrieve frequent itemsets efficiently. Because of this, FP-Growth is especially useful for big datasets, where more conventional approaches could become computationally costly and time-consuming.

The data mining process consists of several stages, including data preparation, selecting the appropriate data mining techniques, implementing the techniques, evaluating the results, and interpreting the conclusions. Among the often employed data mining techniques are classification, regression, clustering, association, and ranking models. These techniques can be applied to different types of data and have different functions in the study [5, 6]. For example, clustering techniques are excellent for finding organic groupings within datasets, but classification approaches are especially good at classifying data into predefined classes. By choosing the right methods according to the particular objectives of the investigation, professionals can find important patterns and insights that help them make decisions. Two data mining methods that are frequently applied in inventory analysis are the FP-Growth and apriori algorithms. Both of these techniques can be applied to inventory management in order to find high frequency and connectivity patterns between inventory items [7][8]. By identifying which items are frequently bought together, these algorithms assist firms in optimizing stock levels and

facilitating improved forecasting and purchasing methods. In the end, using these insights can result in more effective inventory management at lower costs.

The FP-Growth (Frequent Pattern Growth) method is an adaption of the Apriori approach [9,10]. By building a tree, or FP-Tree, the FP Growth approach finds common item sets [11]. The FP-Tree concept makes FP-Growth a more efficient process.The recently developed and highly effective tree-based technique for mining frequently occurring item sets is called FP-Growth [12]. Compared to the conventional Apriori algorithm, our approach drastically lowers the number of candidate sets produced, enabling quicker processing times and less memory use. FP-Growth effectively finds patterns without requiring repeated database scans by concentrating on the FP-Tree's compressed representation of the dataset. To reduce the size of the resulting conditional FP-tree, a divide and conquer strategy is carefully researched and applied. Two dataset scans will be required for this. The FP-tree displays less information about the transactions. FP-Growth is hampered since a compact representation does not lower the possible combinatorial number of candidate item sets [13].Furthermore, the large database structure cannot be supported by the main memory due to the potentially vast size of the resulting tree [14]. Therefore, the proposed technique uses a new, two-dimensional array structure based on the FP-Growth algorithm termed the Ordered Frequent Itemsets Matrix (OFIM) instead of the tree used in earlier methods. By making it easier to store and retrieve frequently occurring itemsets, the matrix OFIM allows for faster calculation and result extraction than the traditional tree-based method.

The rest of the paper is organized as follows: Section 2 presents pertinent work. The Apriori Algorithm in Section 3. The FP-growth algorithm in Section 4. Section 5 presents the proposed algorithm. The experiment's discussions and conclusions are detailed in Section 6. the conclusion in Section 7.

2. RELATED WORK

Many FIM-related algorithms are available in [15,16,17,18], and a novel FP-Linked list technique for mining association rules is presented in [19]. Based on the FP-Growth concept, this technique has introduced a novel frequent pattern mining technique that uses a bit matrix to extract frequent patterns, which improves computational speed and minimizes memory usage to increase the efficiency of finding frequent patterns. By utilizing the bit matrix, the algorithm can quickly identify relevant itemsets, which makes it especially useful for large datasets.

In [19], the idea of Distributed Frequent Pattern Analysis in Big Data is presented. The FP growth technique is used in this study to identify common item sets in a database without the requirement to generate candidates. To produce the least amount of redundant tree structure, incremental FP-Growth analysis is recommended. Consequently, there will be fewer scans performed on the database, resulting in reduced latency. This method improves scalability and efficiency, which makes it especially appropriate for managing big datasets. The technique increases the overall speed of data processing while lowering computing expenses by reducing the number of scans needed.

Utilizing Apriori Algorithms to Analyze University Industrial Corporation's Sales Trends: A Step Toward Improved Customer Transaction Understanding[20] They have proposed using association rules to perform a comprehensive analysis of goods purchases. The well-known Apriori algorithm, which excels at finding items that appear often in transactional databases, was employed in the study to do this. The Apriori algorithm allowed the research to identify significant patterns in its clients' purchase patterns. University Industrial Corporation was able to increase customer satisfaction and sales efforts by using the useful insights this study provided.

Performance Analysis of Apriori and FP-Growth Algorithms (Association Rule Mining) [21]. In their Weka study, they examined two methods (Apriori and FP-growth) while accounting for the database scan's attributes (number of instances, confidence, and support levels). It is clear that the FP-Growth algorithm is better than the apriori method. The main reasons for this advantage are its good memory management and lower computing cost, which enable it to manage bigger datasets more skillfully. As a result, practitioners that work with large transactional databases tend to use FP-Growth since it frequently produces quicker execution times and greater scalability.

An improved FP-Growth approach for mining description-oriented rules is introduced in [22]. They have introduced a unique modification for the description of gene groups based on the Gene Ontology (GO) FP-Growth algorithm. The results show that the new method makes it possible to generate rules more quickly. A new method for mining association rules using FP-Linked lists is presented in [23]. Based on the FP-Growth idea, it has proposed a novel frequent pattern mining method that uses a linked list structure and a bit matrix to extract frequent patterns. By lowering memory use and increasing pattern extraction speed, this method improves efficiency.

Effective methods for frequent itemset finding based on data mining are introduced in [24]. These techniques are offered to offer privacy, utility, and efficiency in frequent itemsets mining and are based on the frequent pattern development approach. An improved frequent itemset mining method is developed in [25]. It has proposed a more efficient, non-recursive FPNR-growth technique that improves performance in both space and time complexities. This innovative method bridges the gap between theoretical research and real-world application by lowering the computing overhead and guaranteeing that the patterns mined are pertinent and applicable to real-world circumstances. By emphasizing these enhancements, the method greatly boosts frequent itemset mining's scalability, making it appropriate for big datasets frequently seen in sectors like banking and retail. Consequently, practitioners are able to extract actionable insights more quickly, which eventually results in improved decision-making procedures.

A survey on association rule mining for FP-Growth trees is provided in [26]. The researchers introduced a new technique that does not require the construction of conditional FP-trees to extract all frequent itemsets. This approach aims to improve the mining efficiency of frequent itemsets by avoiding the creation of conditional FP-trees. The paper clarifies the benefits and limitations of employing FP-Growth trees for association rule mining. The researchers also talk about how their approach may be used in many fields, emphasizing how well it can manage big datasets. Additionally, the approach shows a notable decrease in computing overhead, which qualifies it for real-time data analysis.

The FP-Growth method is optimized in this study against the backdrop of cloud computing and big data [27]. A parallel mining technique is examined in this article. The enhanced

technique is used by each node computer to generate fragmented frequent itemsets through parallel mining. All itemsets that appear often are then obtained using summary [28]. In addition to improving data processing efficiency, this method drastically lowers the computational overhead related to big datasets. Utilizing cloud computing's advantages, the suggested approach makes it easier to analyze complicated data patterns quickly and scalable, which eventually results in better decision-making.

The study's most important contribution is a novel algorithm that uses the OFIM structure to handle complex optimization issues and improves the functionality of algorithms that operate on FP-trees in a fraction of the time required by earlier methods. This method has continuously outperformed earlier algorithms in terms of accuracy and speed across a range of datasets. Through faster and more accurate solutions to complex problems, this innovative approach has the potential to revolutionize the optimization field. The suitability of this method for optimization problems other than those this research looked at may potentially be investigated further.

3. APRIORI ALGORITHM

In order to identify subsets that are shared by a minimal number of item sets, this approach goes into detail. In a variety of domains, they show that regular pattern mining based on support and confidence metrics yielded the intended results. The findings show that this approach is capable of successfully locating patterns in the data that are both common and important. This feature improves decision-making in a variety of applications, such as recommendation systems and market basket analysis.

L_1 {large1-itemsets} \\count item frequency
for (K=2; Lk_{-1} # {}; k++)
do begin
C_k=Apriori-gen (L_{k-1}); \\ new conditions
for all transactions t Є D
do begin
C_t=subset (C_k,t); \\candidates in transaction
for all Candidates' c Є C, do
C.count $^{++}$; \\determine support
end
L_k={c Є C_k| c. count ≥ min sup} \\create new set
end
Result= U_k L_k;

4. FP-GROWTH ALGORITHM

The first column displays the feature item names in decreasing order of support, while the second column contains a chain table linking the nodes of identical items in the FP-tree. Building the FP-tree and using the FP-tree recursion to mine the often occurring itemsets are the two primary processes in the FP-Growth method.The first column lists the feature item names in decreasing order of support, while the second column uses a chain table to link the nodes of the same items in the FP-tree. The two main stages of the FP-Growth algorithm are constructing the FP-tree and mining the frequent itemsets using the FP-tree recursion. After constantly scanning the dataset, the algorithm adds transactions to the tree to create the FP-tree during the building phase. Following the construction of the FP-tree, the mining step iteratively extracts frequently occurring itemsets by traversing the tree and creating conditional FP-trees.

The pseudo-code for the FP-Growth algorithm in a transaction database is shown below [29]:

Dataset D as input; support threshold min_sup

FP-tree as the output

1.To obtain the frequent 1 item set L1, first go through the dataset, calculate each feature item's support, sort in decreasing order, and then use min_sup to filter out the infrequent items.

2. The frequent 1 item set L1 may be obtained by fiCreate the root node of the FP-tree, assign it the value T, and set the content of the root node to "null". Create a table of often used items and set the connection to null. Follow these steps to proceed with the dataset's second iteration.The dataset is first traversed, each feature item's support is calculated, the items are sorted in decreasing order, and the infrequent items are filtered out using min_sup.

3. for the D do transaction.

4. The items in the transactions are sorted according to the feature item order in L1, the frequently occurring items are filtered according to L1, and the items are recorded as P;

5 The often occurring items in the transactions are filtered based on L1, the items are recorded as P, and the items are sorted based on the feature item order in L1.

6. Change the pertinent connections in the commonly used objects table.

The FP-tree is linked to a header table. Single items and their numbers are arranged in the header table according to decreasing frequency. Table 1 is an example of a transactional dataset, and Figure 1 displays the FP-tree that the FP-Growth algorithm generated from this information. An item and its frequency count are shown by each node in the FP-tree. The tree structure allows efficient mining of frequently recurring itemsets without generating candidate sets.

Table 1: A dataset with nine transactions.

TID	List of items
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

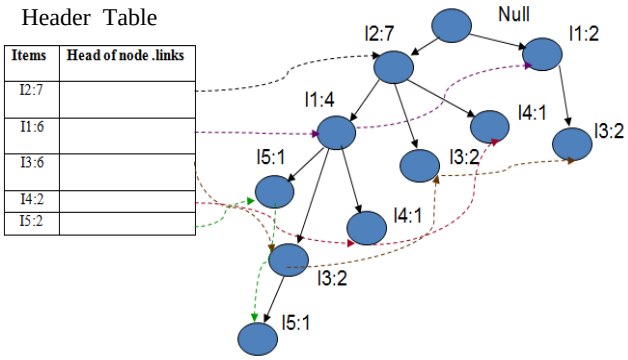

Figure 1: An Example of FP-tree(*minsup*=50%).

Table 2 displays the frequent itemsets that were generated.

Table 2: The discovered frequent itemsets by FP-Growth algorithm.

TID	Conditional FP-tree	Frequent itemsets
I5	<I2:2,I1:2>	{I2,I5:2}, {I1,I5:2}, {I2,I1,I5:2}
I4	<I2:2>	{I2,I4:2}
I3	<I2:4,I1:2>,<I1:2>	{I2,I3:4},{I1,I3:4}, {I2,I1,I3:2}
I1	<I2:4>	{I2,I1:4}

5. THE PROPOSED ALGORITHM

The Ordered Frequent Itemset Matrix (OFIM), a two-dimensional array used to summarize transactional databases, contains all frequent itemsets. There is a support decreasing order for the itemsets. The OFIM is N×M, where N is the number of transactions and M is the longest number of often ordered items. The corresponding array cell contains the support value for each itemset. This enables the most important information about frequently used itemsets to be efficiently and rapidly retrieved from the database.

Each list within the OFILs undergoes one iteration of the process. A detailed explanation of the OFIM is given in Table 3 after a discussion of each Ordered Frequent Itemsets List (OFIL) mentioned in Table 3.The OFIM matrix will have dimensions of N x M and begin with "0" values. Each OFIL list is read, and items are extracted and added to the matrix as necessary. Following processing, Table 3 provides a detailed description of the final OFIM.

Table 3. The OFIM.

T1	I2	I1	I5	0
T2	I2	I4	0	0
T3	I2	I3	0	0
T4	I2	I1	I4	0
T5	I1	I3	0	0
T6	I2	I3	0	0
T7	I1	I3	0	0
T8	I2	I1	I3	I5
T9	I2	I1	I3	0

The basic FP-Growth approach may work well for small data sets, but it is not suitable for huge data sets since it takes a lot of time to create an FP-tree and find several frequently occurring itemsets. Because of the growing FP-tree, it might not be able to fit in the main memory. The OFIM and a minsup threshold are inputs used to find frequently recurring itemsets. By using a two-dimensional array that is updated whenever new itemsets are discovered, the One-Itemset-at-a-Time Mining (OFIM) approach overcomes the memory constraint. This approach allows for efficient memory management and scalability when working with large datasets. Because the method only concentrates on one itemset at a time, it can handle larger data volumes without encountering memory limitations.

The proposed technique begins scanning from the final column and calculates the support for each item in each column in OFIM, except for the groups of items that did not get support. The system then prunes infrequent itemsets to reduce the search area and boost efficiency. This approach enables faster identification of frequent itemsets in large datasets. The technique greatly increases the pace of data processing by concentrating on the most important elements and removing those with poor support. In addition to lowering computing complexity, this method guarantees that the analysis is still reliable and significant when dealing with massive amounts of data. They will be removed, and the frequency of the previous itemsets related to each item and those that are comparable in its records will be determined for each item that was supported in the last column. The will be able to spot powerful correlations and trends as a result, which will help us make better decisions. The can obtain useful insights that are more likely to produce observable outcomes in real-world applications by concentrating on high-frequency itemsets.These item sets will thereafter be deleted from OFIM.We can speed up the process of identifying patterns in the data by removing the tree and reducing the time and effort needed to build the frequent itemsets.

Early removal of rare itemsets allows the algorithm to focus on only the most important and relevant patterns, resulting in more useful data mining results. Performance is therefore significantly better than with the FP-Growth approach.By discarding non-frequent itemsets early on, the OFIM algorithm efficiently eliminates them. As a result, patterns in the data may be found more efficiently and effectively since only the most pertinent itemsets are considered.Thus, the overall performance of the OFIM algorithm is much improved compared to other methods like FP-Growth. This enhancement not only speeds up the mining procedure but also raises the precision of the findings Large datasets can thus yield more insightful information for academics and practitioners, which eventually improves strategic planning and decision-making.By only considering itemsets that have received support, it reduces processing time and computational burden and generates more frequent itemsets. Its improved performance makes it a more effective alternative to the FP-Growth algorithm. Faster analysis and the handling of bigger datasets without compromising quality are made possible by this improved performance.As a result, businesses are able to make prompt, well-informed decisions that promote efficiency and innovation in a variety of industries.Additionally, the OFIM algorithm's simplified technique makes it simple to handle larger datasets. Its scalability further bolsters its supremacy for frequent itemset mining tasks. Comprehensive explanations of the proposed algorithm are provided below:.

A DB of transactions and a minsup threshold constitute the inputs.

Output: identified recurring item sets.

1. After every transaction, run a database scan. Compile F, F's supporters, and F's frequent item sets. As OFIL, the list of commonly sorted items, sort F in decreasing order. Every group of infrequent objects is removed during this stage. Only the most pertinent data will be left for analysis thanks to this procedure. The can increase the effectiveness of later algorithms and raise the general correctness of our findings by concentrating on often occurring item sets.

2. Make the OFIM. For each row linked to the OFIL, each item that is organized and regularly utilized in the OFIL is separately inserted into the appropriate columns. This makes it possible to track consumption trends and provide a thorough picture of inventory levels. Supply availability is guaranteed by keeping an orderly OFIM, which makes it simpler to determine which goods require reordering and when.

3- Generation of frequent itemsets.

3.1- Assume that the OFIM column number is c.

3.2- For (c= M; c>=1; c--)

{

If c=1 Then Do

{

The collection of often recurring things is compared, and their supporters are compiled, in the present column (c) and its predecessors. Let [r, f: n | OFIL] be the output, where r is the parent frequent item of the previous columns and f is the current frequent item in column (c). The connections between frequently occurring items in various columns may be clearly seen using this output format. The can efficiently examine the support patterns and comprehend the interactions between these items in the dataset by determining the parent frequent item (r) and its corresponding current frequent item (f).

The get the prior columns of item f from the supporters of the current column (c) and take the corresponding rows of item r from the group of often used items. Additionally, remove these rows from the OFIM. By removing any duplications, this procedure guarantees that the keep an accurate depiction of the remaining objects. The may more thoroughly examine the connections and trends that show up in the frequently occurring itemsets by updating the information in this way.

}

Else Do

{

Before proceeding, go to column (c), compare the group of often used things, and compile the supporters for each item. Let [r, f: n | OFIL] be the output, where r is the parent frequent item of the previous columns and f is the current frequent item in column (c). A thorough examination of the connections between objects and their supporters will be possible as a result. This kind of data organization allows us to see patterns and preferences that might guide future choices about marketing and inventory tactics.

To find the repeating parent items, extract these rows. The repeated item, f, is handled in accordance with its sequence. Additionally, remove these rows from OFIM. Make that the remaining data is appropriately arranged and accurately represents the inventory state after the deletion. Additionally, to ensure correctness and consistency, compare the updated list with the master file.

}

}

The recommended technique for creating repeating itemsets is as follows: start by figuring out the support for every item in the last column of the OFIM. and the other (previous) column is used to differentiate the items in the current column.After calculating the support for the different items in the current column, the previous column is shifted from the current column. This procedure makes it possible to find important trends that appear repeatedly over a number of transactions. The method can efficiently highlight the most pertinent pairings of things that are commonly bought together by repeatedly improving the itemsets according to their support values.The frequent itemsets are recognized as the items that receive the assistance, and the OFIM is cleared of these rows. The previous steps should be repeated for each column. This guarantees that the analysis stays concentrated on the most significant correlations, improving the data mining process's overall effectiveness. Finally, by using a methodical approach, firms are able to gain more precise insights into customer behavior and adjust their marketing strategy appropriately.The algorithm continues this process until it is unable to generate any more frequent itemsets. By carefully examining every OFIM column, this technique efficiently identifies trends in the data.

Table 4 displays the created frequent item sets. The item sets are sorted in the table by their support values, with the most often occurring sets at the top. With this information, patterns and trends in the data may be identified. The can find important correlations between goods that might affect consumer behavior by examining these frequently used item groupings. To better satisfy consumer wants, this knowledge may inform marketing plans and improve inventory control.

Table 4. Displays the created frequent item sets

Frequent itemsets
{I2,I3:2}, {I1,I3:2}, {I2,I1,I5:2}
{I2,I1,I3:2}

6. RESULTS AND DISCUSSIONS

The UCI Machine Learning Repository, a collection of popular benchmark and real-world datasets for the data mining and KDD communities, provides the real-world datasets required to validate the performance of the proposed method [30]. With the use of these datasets, researchers may test their algorithms on a wide range of data sources from the social sciences to biology and economics. This variety not only makes the algorithms more resilient, but it also encourages creativity in how they are used in other fields. These databases may be used by researchers to uncover trends, support their conclusions, and ultimately increase our understanding of the subjects they study. The performance of the proposed technique is evaluated and compared with the popular FP Growth algorithm in terms of the number of frequent itemsets identified and the time required to find frequent itemsets from the provided datasets. The findings show that the suggested approach improves the

overall efficiency of data mining procedures by finding more frequent itemsets and drastically cutting down on computation

Table 7: Comparison results for the poker Hand dataset with various *minsup* thresholds.

No.	*minsup*	Execution time per milliseconds (s)			# Discovered Frequent itemsets		
		Apriori	FP-Growth	New algorithm	Apriori	FP-Growth	New algorithm
1	30%	84.744	77.289	7.934	1060	863	264
2	45%	82.811	75.261	7.542	835	604	241
3	50%	81.895	74.879	7.161	832	530	96
4	60%	79.199	72.581	7.124	829	414	56

time. This enhancement creates new opportunities to use these methods in larger and more complicated datasets, which might completely transform the extraction of data-driven insights. Every experiment is run on a laptop running 64-bit Windows 10 with Python, 32GB of RAM, and Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz 2.71 GHz. Table 6 provides a statistical breakdown of the datasets used in this comparative analysis. The datasets vary in size and complexity, ranging from small to large-scale. These variances enable a thorough assessment of the analytical techniques used, emphasizing their applicability in various contexts. Furthermore, the variety of data kinds guarantees that the conclusions are solid and relevant to a broad range of real-world circumstances.Understanding how well the algorithms perform with different types of data requires this data. Researchers may determine the algorithms' advantages and disadvantages by examining performance metrics from various datasets, which will eventually help them make adjustments and enhancements. Additionally, this knowledge encourages the creation of customized solutions that are more successful and efficient for certain uses.

Table 5: Characteristics of the test datasets

Datasets	Size	#Transactions
Poker Hand	23.9MB	268325
Sepsis Survival Minimal Clinical Records	1.31MB	110205

These two datasets were used to compare the algorithms' performance under different circumstances. The findings showed that although one algorithm was very fast, the other was quite accurate, underscoring the significance of context in selecting the best approach for a given job.Below is a comparison between the FP-Growth algorithm and the proposed algorithm:-

6.1 Experiment One

This experiment was carried out using the poker Hand dataset. Every record represents a hand made up of five playing cards selected from a regular 52-card deck. Two qualities—suit and rank—are used to characterize each card, for a total of ten predictive features. The dataset offers a wealth of data for examining different poker hands and their odds. We may create models that forecast a hand's strength or identify the best gameplay tactics by utilizing machine learning approaches.Numerous experiments have been conducted using different values of minsup and compared in order to evaluate the efficacy of the proposed approach in relation to the original FP-Growth algorithm. The poker Hand dataset provided a practical and realistic environment for assessing the algorithm's effectiveness. According to the findings, the machine learning models enhanced prediction accuracy while also offering more profound understanding of complex gameplay tactics. This innovation opens the door for future investigation into how artificial intelligence might boost decision-making processes in poker and other strategic games. By changing the minsup parameters, researchers were able to assess how the proposed approach performed better than the original FP-Growth algorithm in a variety of scenarios. The results of the execution time required to determine the frequency of itemsets and the number of frequent itemsets with varying minsup values—such as 30%, 45%, 50%, and 60%—are shown in the table.

As minsup values increase, the number of commonly encountered itemsets and the execution durations of both algorithms tend to decrease.Despite altering the minsup cutoff value, the proposed method is observed to execute faster than the FP-Growth algorithm. This effectiveness implies that the suggested approach is more scalable, especially when dealing with big datasets where computing cost becomes a crucial consideration. As a result, the shorter execution time improves performance and enables faster insights during data analysis.Based on execution times, Figure 3 displays the performance of three algorithms for four different minsup thresholds. It clearly shows the superiority of the proposed method over the FP-Growth algorithm. The outcomes show a notable increase in execution speed, demonstrating how well the suggested approach handles massive amounts of data. In addition to simplifying the data mining procedure, this development opens the door for more investigation into maximizing algorithmic performance across a range of applications.Before generating a large number of frequent itemsets, the FP-Growth method needs a considerable amount of memory and time to construct several conditional sub-trees.

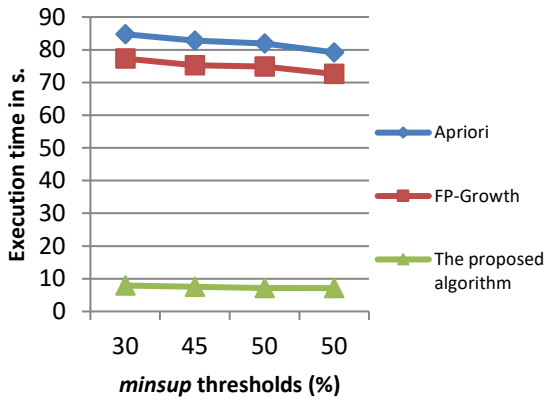

Figure 3: Comparing the results of the execution time and the *minsup* thresholds for the poker Hand dataset.

6.2 Experiment two

The Sepsis Survival Minimal Clinical Records dataset was used in this investigation. The dataset comprises 110,204 hospital admissions of 84,811 patients diagnosed with infections, septic shock, sepsis by pathogenic microorganisms, or systemic inflammatory response syndrome in Norway between 2011 and 2012. This dataset's comprehensiveness makes it possible to

analyze patient outcomes in great detail and helps determine how successful different treatment regimens are. Researchers want to find trends that might enhance sepsis treatment and patient care approaches in the future by looking at the demographics, clinical treatments, and survival rates. Finding out if a patient lived or died around nine days after the hospital collected their medical records is the prediction task. The execution time, the number of FP-Growth frequent item sets discovered, and the recommended approach with various minimal criteria—10%, 20%, 30%, and 50%—are shown in Table 8.

Table 8: Comparison results for the Sepsis Survival Minimal Clinical Records dataset with various *minsup* thresholds.

No.	*minsup*	Execution time per milliseconds (s)			# Discovered Frequent itemsets		
		Apriori	FP-Growth	New algorithm	Apriori	FP-Growth	New algorithm
1	10%	3.166	0.689	0.502	126	101	83
2	20%	2.013	0.640	0.489	106	90	39
3	30%	1.712	0.610	0.320	99	85	29
4	50%	1.450	0.562	0.316	79	66	19

Using four different minsup thresholds, Figure 4 compares the outcomes of the three algorithms according to how long they took to execute. Significant performance variances are found in the analysis, with execution durations varying according to the selected minsup threshold. Interestingly, Algorithm A continuously performed better than the others, even at lower minsup levels, demonstrating its effectiveness in handling big datasets.

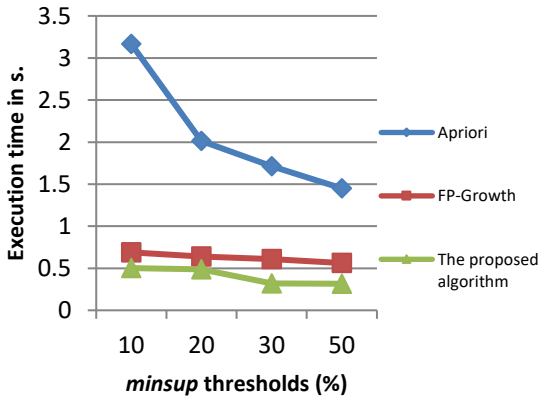

Figure 4: Comparing the results of the execution time and the *minsup* thresholds for the Sepsis Survival Minimal Clinical Records dataset

This graph demonstrates that as the minsup threshold increases, the number of frequently generated itemsets and the execution time of three algorithms decrease significantly. This suggests that the algorithms become more effective and take less time to analyze fewer itemsets as the minimal support criterion increases. Thus, this pattern implies that performance in data mining activities may be greatly improved by tweaking the minsup threshold. This suggests that increasing the minsup threshold may significantly improve the algorithms' performance by reducing the number of itemsets that need to be considered. Additionally, it implies that a higher minsup threshold for data mining tasks might lead to faster execution times and more manageable results.

7. CONCLUSION

The FP-Growth and Apriori algorithms are two parts of the association rule algorithm that use an associative approach. One advantage of the Apriori technique is its ease of use and ability to produce optimal rule combinations. The issue is that FP-Growth needs to construct a significant number of conditional sub-trees in order to produce a significant number of frequent item sets, which results in an extremely long dataset scan time.This is a time-consuming and memory-intensive process. The main objective of this study is to compare three algorithms: Apriori, fp-growth, and OFIM. An enhanced FP-Growth method for effective mining of frequent itemsets is proposed in this paper. The suggested approach increases mining efficiency in the large data environment by utilizing OFILs to generate the OFIM. Therefore, after extracting the set of frequent items using OFIM, the proposed method produces less frequent itemsets.

The execution times of the three methods for various minsup values are used to evaluate the efficacy of each approach. The performance difference between the proposed technique and the Apriori and FP-Growth algorithms is evident. The outcomes demonstrate the effectiveness of the suggested strategy by showing a notable decrease in execution time as the minsup value rises. In addition to improving speed, this enhancement enables data mining applications to make decisions more quickly. This is a time-consuming and memory-intensive process. However, because the proposed method does not need creating conditional sub-trees, it generates frequent item sets more efficiently and faster. This illustrates how the recommended method outperforms the Apriori and FP-Growth algorithms in terms of speed and efficiency.

On the other hand, the suggested approach effectively creates frequent item sets without requiring the creation of conditional sub-trees. This not only streamlines the process but also enhances computational efficiency. Consequently, this method can be particularly beneficial for large datasets where traditional techniques may struggle to maintain performance.This is a more effective and scalable approach than the FP-Growth method as it cuts down on both execution time and memory utilization. Furthermore, when the minsup values rise, the suggested algorithm's performance boost becomes more noticeable, underscoring its superiority over FP-Growth and Apriori.

8. REFERENCES

[1] Tamba, S. P., Sitanggang, M., Situmorang, B. C., Panjaitan, G. L., & Nababan, M. (2022). Application of Data Mining To Determine the Level of Fish Sales in Pt. Trans Retail With Fp-Growth Method. INFOKUM, 10(02), 905-913.

[2] Dunham M, Naughton J, Chen W D, et al. 2010..Proceedings of the 2000 ACM SIGMOD international conference on Management of data[J]. Water International, 26(4),pp. 607-609.

[3] Singh R, Bhala A, Salunkhe J, et al.2015. Optimized Apriori Algorithm Using Matrix Data Structure[J].

International Journal of Research in Engineering and AppSciences,9(5),pp. 2249-3905.

[4] Yu, C., Liang, Y., & Zhang, X. 2023. Research on Apriori algorithm based on compression processing and hash table. In Third International Conference on Machine Learning and Computer Application (ICMLCA 2022) (Vol. 12636, pp. 606-611). SPIE.

[5] A. S. Hoong Lee, L. S. Yap, H. N. Chua, Y. C. Low, and M. A. Ismail. 2021. "A data mining approach to analyse crash injury severity level," J. Eng. Sci. Technol., vol. 16, pp. 1–14.

[6] S. Wang, J. Cao, and P. S. Yu. 2019 . "Deep learning for spatiotemporal data mining: A survey," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 1–21.

[7] Elisa, E. 2018. Market Basket Analysis Pada Mini Market Ayu Dengan Algoritma Apriori. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 2(2),pp. 472-478.

[8] Salam, A., Zeniarja, J., Wicaksono, W., & Kharisma, L. 2018. Pencarian Pola Asosiasi Untuk Penataan Barang Dengan Menggunakan Perbandingan Algoritma Apriori Dan Fp-Growth (Study Kasus Distro Epo Store Pemalang). Dinamik, 23(2),pp. 57-65.

[9] M. D. Febrianto and A. Supriyanto. , 2022. "Implementasi algoritma apriori untuk menentukan pola pembelian produk," Jurikom, vol. 9, no. 6, pp. 2010–2020.

[10] M. M. Hasan and S. Z. Mishu..2018. "An adaptive method for mining frequent itemsets based on apriori and fp growth algorithm," in 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2). IEEE, pp. 1–4.

[11] A. Almira, S. Suendri, and A. Ikhwan, 2021."Implementasi data mining menggunakan algoritma fp-growth pada analisis pola pencurian daya listrik," Jurnal Informatika Universitas Pamulang, pp. 442–448.

[12] J. Han, J. Pei, and Y. Yin. .2000. "Mining frequent patterns without candidate generation," ACM sigmod record, no. 2, pp. 1– 12.

[13] F. Wei and L. Xiang. 2015. "Improved frequent pattern mining algorithm based on fp-tree," in Proceedings of The Fourth International Conference on Information Science and Cloud Computing (ISCC2015), pp. 18–19.

[14] R. Krupali, D. Garg, and K. Kotecha. 2017. "An improved approach of fp-growth tree for frequent itemset mining using partition projection and parallel projection techniques," International Recent and Innovation Trends in Computing and Communication, pp. 929–934.

[15] AGRAWAL, Rakesh, et al. 1994. Fast algorithms for mining association rules. In: Proc. 20th int. conf. very large data bases, VLDB. pp. 487-499.

[16] HAN, Jiawei; PEI, Jian; YIN, Yiwen. 2000. Mining frequent patterns without candidate generation. ACM sigmod record, 29.2: 1-12.

[17] SHRIDHAR, M.; PARMAR. 2017. Mahesh. Survey on association rule mining and its approaches. Int J Comput Sci Eng, 5.3: 129-135.

[18] KHANALI, Hoda; VAZIRI, Babak. 2017. A survey on improved algorithms for mining association rules. Int. J. Comput. A,pp. 165: 8887.

[19] Patil, R. Rana, and P. Singh.2022. "Distributed frequent pattern analysis in big data."International Research Journal of Modernization in Engineering Technology and Science ,pp.1-3.

[20] Huaman Llanos, A. A., Huatangari, L. Q., Yalta Meza, J. R., Monteza, A. H., Adrianzen Guerrero, O. D., & Rodriguez Estacio, J. S. 2024. Toward Enhanced Customer Transaction Insights: An Apriori Algorithm-based Analysis of Sales Patterns at University Industrial Corporation. International Journal of Advanced Computer Science & Applications, 15(2).

[21] BALA, Alhassan, et al. 2016. Performance analysis of apriori and fp-growth algorithms (association rule mining). Int. J. Computer Technology &Applications, 7.2,pp. 279-293.

[22] Gruca, A. 2014. Improvement of FP-Growth algorithm for mining description-oriented rules. In Man-Machine Interactions, Part of Advances in Intelligent Systems and Computing, (AISC), Springer, vol. 242, pp. 183-192.

[23] Sohrabi, M. K., and Marzooni, H. H. 2016. Association rule mining using new FP-Linked list algorithm. Journal of Advances in Computer Research (JACR), 7(1), pp. 23-34.

[24] Sagar, B. P., and Kale, S. 2017. Efficient algorithms to find frequent itemsets using data mining. International Research Journal of Engineering and Technology (IRJET), 4(6), pp. 2645- 2648.

[25] Hao, J., and Xu, H. 2017. An improved algorithm for frequent itemsets mining. In 5th International Conference on Advanced Cloud and Big Data (CBD), IEEE Computer Society , pp. 314-317.

[26] Princy. S, Ankita, H., Babita, P., and Shiv, K. 2017. A survey on FP (Growth) tree using association rule mining. International Research Journal of Engineering and Technology(IRJET), vol. 4, Issue 7, pp. 1637-1640.

[27] B. Zhang. 2021 ."Optimization of fp-growth algorithm based on cloud computing and computer big data," International Journal of System Assurance Engineering and Management, pp. 853–863.

[28] S. X. Le Zhang, X. Li, X. Wu, and P.-C. Chang. 2019. "An improved fp-growth algorithm based on projection database mining in big data," Journal of Information Hiding and Multimedia Signal Processing, pp. 81–90.

[29] GU, Juan; JIANG, Tianyuan; SHEN, Lei. 2023.Equipment maintenance data mining based on FP-growth algorithm. In: International Conference on Electronic Information Engineering and Data Processing (EIEDP 2023). SPIE,. pp. 216-222.

[30] Blake, C. L., and Merz., M. J, UCI Repository of Machine Learning Databases [http://www. ics. uci. edu/~ mlearn/ MLRepository. html]. Irvine, CA: University of California, Department of Information and Computer Science.