ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

Enhanced FP-Growth Framework and Apriori Algorithm Utilizing TDA for Big Data Analysis

Abdulkader Mohammed Abdulla Al-Badani*[1]

[1] Faculty of Science and Engineering,
Department of Computers,, Aljazeera University, Ibb, Yemen.

Publication Date: 2025/12/19

Abstract: To efficiently analyze huge datasets, mining big data requires advanced computational techniques and algorithms. Apriori and FP-Growth are two of the most well-known algorithms in data mining. They help businesses make decisions based on customer trends and behaviors by finding patterns and correlations. Machine learning has made these algorithms even better by making them more accurate and efficient. The association rule approach does have some problems, though. For example, it needs a lot of memory, it has to search through all the data sets to find the frequency of an item set, and it sometimes makes rules that aren't the best. This study conducts a comparative analysis of the FP-Growth, Apriori, and TDA algorithms, demonstrating notable performance differences. The FP-Growth algorithm was much better at working with large datasets than the Apriori method, which had problems with scalability and took longer to process larger datasets, even though it was easier to build. This study suggests changes to the FP-Growth algorithm to fix these problems. It uses the TDA matrix to make a very compact FP-tree. This method tries to cut down on the time it takes to mine and the number of items that are created, which will make memory use more efficient and speed up processing for large datasets. In short, the proposed method is a promising way to make data mining processes more efficient and scalable, especially when it comes to big data analytics.

Keywords: *FP-Growth Algorithm, Aprioiri Algorithm, FP-tree, Support Count, TDA.*

How to Cite: Abdulkader Mohammed Abdulla Al-Badani (2025) Enhanced FP-Growth Framework and Apriori Algorithm Utilizing TDA for Big Data Analysis. *International Journal of Innovative Science and Research Technology*, 10(12), 919-928. https://doi.org/10.38124/ijisrt/25dec579

I. INTRODUCTION

In today's world of information technology, the rapid increase in data creation means that we need better ways to collect and store data. This change has greatly improved the ability to collect and store huge amounts of data[1], which makes it easier to analyze all of that data and lets businesses get useful information from large amounts of it. Being able to handle and understand large datasets is now an important part of making decisions in many fields. Association rule mining is one of the many methods that have been created to deal with these problems. It is a key way to find patterns in large datasets.

Association rule mining, especially the Apriori algorithm, has been very helpful in finding frequent itemsets and making Boolean association rules[2]. Since it was first created, the Apriori algorithm has been improved many times, making association analysis faster and more accurate. Even with these improvements, the algorithm's dependence on candidate generation and multiple database scans makes it hard to work with large datasets quickly. To deal with these problems, new methods like the FP-Growth algorithm have come up. The FP-Growth algorithm uses a tree structure to make itemsets, which cuts down on the number of times the database needs to be scanned for association analysis.

Even though the FP-Growth algorithm has come a long way, we still need to improve data mining methods so that we can get more information from large datasets. Current research is still looking into new ways to make data mining processes more efficient, with the goal of getting around current problems and making them work better. This research aims to augment the existing discourse by exploring innovative methodologies that improve the efficiency and efficacy of association rule mining[3]. This research seeks to enhance the field of data mining by filling existing knowledge gaps and offering valuable insights for businesses aiming to utilize large datasets for strategic decision-making.

The field of data mining has become more and more important for getting useful information from large datasets[4]. This helps people make decisions in many different areas. Data mining involves a number of important steps, including preparing the data, choosing the right methods, putting them into action, looking at the results, and figuring out what they mean. Every step is important for making sure that the results are correct and useful.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

Classification, regression, clustering, association, and ranking models are some of the most common data mining methods because they can handle many different types of data and are very useful [5, 6]. Clustering techniques are especially good at finding natural groups in datasets, while classification methods are great at putting data into groups that have already been set up. Researchers can find important patterns and insights that can help them make smart decisions by carefully choosing these methods based on the study's goals.

Even though data mining techniques have come a long way, they are still not widely used in inventory management, especially when it comes to improving stock levels and making better predictions. Analyzing inventory is an important part of running a business. Good management can cut costs and improve service delivery. FP-Growth and Apriori are two well-known data mining algorithms that have shown promise in this area. These algorithms are very helpful for finding patterns in inventory items that happen often and connect with each other [7, 8]. They show businesses which items are often bought together, which lets them improve their stock levels and buying strategies, which leads to better inventory management.

The FP-Growth (Frequent Pattern Growth) method is a modified version of the Apriori method [9,10]. The FP Growth method finds common sets of items by making a tree, or FP-Tree [11]. FP-Growth works better because of the FP-Tree idea. FP-Growth [12] is a new and very effective tree-based method for mining sets of items that happen a lot. Our method makes a lot fewer candidate sets than the standard Apriori algorithm, which speeds up processing and uses less memory. FP-Growth effectively identifies patterns without necessitating multiple database scans by focusing on the compressed representation of the dataset within the FP-Tree. A divide-and-conquer strategy is carefully studied and used to make the conditional FP-tree smaller. You will need to scan the dataset twice for this. The FP-tree shows fewer details about the transactions. FP-Growth is limited because a compact representation does not reduce the potential combinatorial number of candidate item sets [13]. Also, the main memory can't handle the large database structure because the tree that comes from it could be very big [14]. The proposed technique employs a novel two-dimensional array structure derived from the FP-Growth algorithm, referred to as the Two-Dimensional Array (TDA), in lieu of the tree structure utilized in previous methods. The matrix TDA lets you store and get to frequently occurring itemsets more easily, which speeds up calculation and result extraction compared to the old tree-based method.

Here is how the rest of the paper is set up: Section 2 contains relevant work. Section 3 talks about the Research Method. Section 4 has great information about the TDA. The proposed algorithm is shown in Section 5. Section 6 goes into great detail about the experiment's discussions and conclusions. the end of Section 7.

II. RELATED WORK

The literature on Frequent Itemset Mining (FIM) has progressed considerably[15,16,17,18], resulting in the creation of various algorithms aimed at improving the efficiency and scalability of mining association rules. A new FP-Linked list method has been introduced, as explained in [19]. This method is based on the FP-Growth idea. This new method uses a bit matrix to find common patterns, which speeds up calculations and uses less memory. The use of a bit matrix makes it easy to quickly find relevant itemsets, which makes the algorithm especially useful for working with large datasets. This progress meets the urgent need for effective data mining methods that can deal with the growing amount and complexity of data in modern applications.

[16] talks about Distributed Frequent Pattern Analysis in Big Data, which shows that people are still interested in distributed data processing. This study employs the FP-Growth algorithm to discover frequent itemsets without the need for candidate generation. The addition of incremental FP-Growth analysis is especially important because it tries to cut down on unnecessary tree structures and database scans. This method greatly improves scalability and efficiency, making it a great choice for managing large amounts of data. The method speeds up data processing by lowering latency and computational costs by scanning less often.

Even with these improvements, there are still some gaps in our understanding of frequent pattern mining[17]. Furthermore, the amalgamation of these methodologies with nascent technologies like machine learning and artificial intelligence offers a promising direction for future research.

In recent years, there has been a lot of interest in using association rule mining to look at sales patterns, especially to get better insights into customer transactions[20]. The Apriori algorithm, known for its ability to find important patterns in how customers buy things, helped improve customer satisfaction and guided sales efforts. This study showed how useful the algorithm is for getting useful information from transactional data, which helps businesses make better decisions.

Conversely, a comparative performance analysis of the Apriori and FP-Growth algorithms was performed, emphasizing the respective advantages of each method in association rule mining [21]. The study used the Weka platform to look at the algorithms based on different factors, such as the number of instances, confidence, and support levels. The results showed that the FP-Growth algorithm worked better than the Apriori method. This was mostly because it managed memory better and cost less to run. Because of these features, FP-Growth can handle larger datasets more efficiently, which means it runs faster and can grow more easily. As a result, FP-Growth is preferred by professionals who work with large transactional databases because it always provides faster processing and stronger performance.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

In [22], a better FP-Growth method for mining rules that are based on descriptions is introduced. They have made a unique change to how gene groups are described using the Gene Ontology (GO) FP-Growth algorithm. The results show that the new method lets you make rules faster. Reference [23] introduces a new way to mine association rules using FP-Linked lists. It has come up with a new way to mine frequent patterns that uses a linked list structure and a bit matrix to find them. This is based on the FP-Growth idea. This method makes things more efficient by using less memory and speeding up the process of finding patterns.

In [24], you can learn about good ways to find frequent itemsets using data mining. These methods are based on the frequent pattern development approach and are meant to protect privacy, usefulness, and speed in mining frequent itemsets. In [25], a better way to mine frequent itemsets is created. It has put forward a more effective, non-recursive FPNR-growth method that boosts performance in terms of both space and time complexities. This new method closes the gap between theoretical research and real-world use by lowering the computing overhead and making sure that the patterns mined are useful and relevant to real-world situations. By focusing on these improvements, the method makes frequent itemset mining much more scalable, which is useful for large datasets that are common in fields like banking and retail. As a result, practitioners can get useful information faster, which leads to better ways of making decisions in the long run.

The literature on association rule mining has changed a lot over time. FP-Growth trees have become a popular method because they work well with large datasets. A significant research by [26] presents a novel method that eliminates the necessity for building conditional FP-trees, thereby improving the efficiency of frequent itemset mining. This improvement solves a major problem with traditional FP-Growth methods, which often require a lot of extra computing power because conditional tree construction is recursive. By removing this step, the suggested method not only makes mining easier but also makes it useful in more areas that need real-time data analysis.

The study further clarifies the benefits and limitations of FP-Growth trees in association rule mining. It shows how well the method can handle large datasets, which is very important in areas like market basket analysis, bioinformatics, and network traffic analysis. The decrease in computational overhead is especially important because it makes this method a good choice for real-time applications where speed and efficiency are very important.

A major part of this research is the creation of a new algorithm that uses the TDA (Two-Dimensional array) structure. This algorithm solves difficult optimization problems and makes FP-tree-based algorithms work better, getting better accuracy and speed than earlier methods. The fact that it consistently outperforms different datasets shows how strong it is and how it could change the way optimization works. The algorithm's capacity to provide expedited and accurate solutions to complex issues indicates favorable prospects for its utilization in optimization contexts beyond those initially investigated in the research.

III. APRIORI ALGORITHM

The method described in this study successfully identifies subsets that are shared by only a few item sets, showing that it can be used in many different fields. The method consistently produces accurate and meaningful results by using regular pattern mining techniques based on support and confidence metrics. The research results show that the method can successfully find patterns in data that are both common and important. This ability to see important patterns makes it easier to make decisions in many situations, such as recommendation systems and market basket analysis. This shows how useful and flexible the approach is in real life.

In other words, the difference in execution time and magnitude of results between the two databases points out how much the number of attributes and the general operation complexity affect the performance of the algorithm. The underlying message here is that, while one is constructing the algorithm, the database architecture should be understood first because it reflects the efficiency and effectiveness of the data to be processed.

IV. FP-GROWTH ALGORITHM

The FP-Growth algorithm is an important tool for data mining, especially for quickly finding frequent itemsets without having to create candidates. The algorithm has two main steps: building the FP-tree and using FP-tree recursion to find frequent itemsets. To start, the FP-tree is built by scanning the dataset to find feature items. These items are then put in the first column in order of their support, from highest to lowest. This structure makes sure that the most important things come first, which makes it easier to get around. At the same time, the second column has a chain table that connects nodes of the same items in the FP-tree. This keeps the structure consistent, which is important for recursive mining. During the mining phase, the algorithm goes through the FP-tree in a systematic way to find frequent itemsets. It uses conditional FP-trees to narrow down and isolate patterns. This two-phase method not only makes computations more efficient, but it also makes traditional candidate generation methods much less complicated. Because of this, the FP-Growth algorithm is a great way to find patterns in large datasets. It is both fast and accurate for frequent itemset mining.

Below is the pseudo-code for the FP-Growth algorithm in a transaction database [27]:

Input: Dataset D; support threshold min_sup
The output is an FP-tree.

- To get the frequent 1 item set L1, first go through the dataset and find the support for each feature item. Then, sort the items in order of support and use min_sup to get rid of the less common ones.
- You can get the frequent 1 item set L1 by fi Make the root node of the FP-tree, give it the value T, and set the content

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

of the root node to "null." Make a list of items that are often used and leave the connection blank. To move on to the second iteration of the dataset, follow these steps. First, the dataset is traversed, and the support for each feature item is calculated. Then, the items are sorted in order of support, and the infrequent items are filtered out using min_sup.

- for the D to do business.
- The items in the transactions are sorted based on the feature item order in L1, and the items that show up most often are filtered based on L1. The items are then recorded as P.
- The items that show up a lot in the transactions are filtered based on L1, recorded as P, and sorted according to the feature item order in L1.
- Change the connections that are relevant in the table of commonly used objects.

There is a header table that the FP-tree is linked to. The header table sorts single items and their numbers by how often they appear, from most to least. Table 1 is a transactional dataset, and Figure 1 shows the FP-tree that the FP-Growth algorithm made from this data. Every node in the FP-tree shows an item and how many times it appears. The tree structure makes it easy to mine itemsets that come up often without making 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

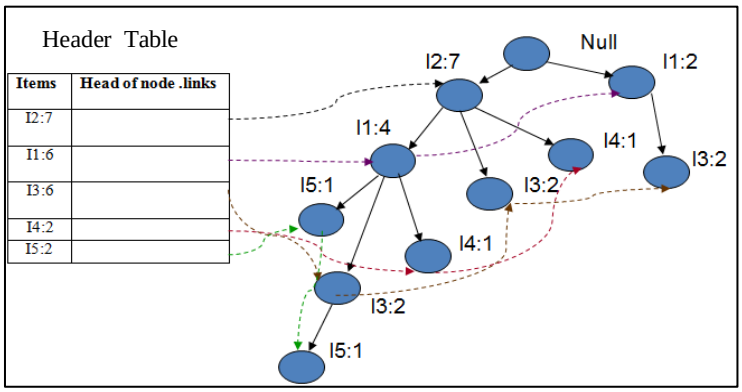

Fig 1: An Example of FP-tree(*minsup*=50%).

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}

V. THE PROPOSED ALGORITHM

The Two-Dimensional Array (TDA) is an important tool for summarizing transactional databases because it organizes all of the frequent itemsets in a way that makes sense. The TDA is organized in a way that makes it easier to find support. Its dimensions are N×M, where N is the number of transactions and M is the maximum number of frequently ordered items. Each cell in this array holds the support value for the itemsets that go with it. This makes it easy and quick to get important information about frequently used itemsets from the database. This systematic arrangement not only makes it easier to get to the data, but it also makes it easier to analyze transactional patterns.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

Table 3 shows how each Ordered Frequent Itemsets List (OFIL) is processed over and over again to build the TDA. At first, the TDA matrix is filled with "0" values, which sets a baseline for adding more data later. During each iteration, items from the OFIL lists are taken out and added to the matrix. This makes the representation of transactional data more accurate. After this process is done, Table 3 gives a full explanation of the finished TDA, showing how well it works to summarize and present transactional insights. This methodical approach shows how useful the TDA is for making data analysis in transactional databases more accurate and efficient.

Table 3. The TDA.

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 FP-Growth algorithm works well with small datasets, but it has big problems when used with large datasets because it needs a lot of memory and takes a long time to build an FP-tree and find frequent itemsets. The FP-tree may grow too big for the main memory, making the method useless for analyzing large amounts of data. The One-Itemset-at-a-Time Mining (TDA) method, along with a minimum support (minsup) threshold, is a good way to deal with these problems. This method uses a two-dimensional array that is updated in real time with new itemsets. This solves the memory problems that come with the traditional FP-Growth method. The TDA method improves memory management and scalability by focusing on one itemset at a time. This makes it possible to process large datasets efficiently without running out of memory. The TDA approach is a big step forward in the field of data mining because it gives you a strong way to work with large amounts of data.

The suggested method starts scanning the dataset from the last column and uses the Transactional Data Analysis (TDA) to figure out how much support each item in each column gets. It then skips groups of items that don't get enough support. The system improves efficiency by cutting down on the search space by removing infrequent itemsets. This method makes it easier to quickly find frequent itemsets in big datasets by focusing on the most important elements and getting rid of those with little support. So, this method not only makes the math easier, but it also makes sure that the analysis is accurate and important when dealing with large amounts of data. The last column will show the frequencies of the previous itemsets and those of similar records. This will help find strong correlations and trends. Such insights are useful for making smart choices because they are more likely to lead to real-world results. The method lets you get useful information by focusing on high-frequency itemsets, which are then taken out of the TDA. This process speeds up the process of finding patterns in the data, which cuts down on the time and effort needed to make frequent itemsets. This makes data analysis easier and more efficient overall.

By getting rid of rare itemsets early on, the TDA algorithm becomes more efficient because it focuses on the most important and relevant patterns. This leads to better data mining results. This method works much better than the FP-Growth method. The TDA algorithm speeds up the mining process by quickly getting rid of non-frequent itemsets. This makes it easier and more accurate to find patterns in the data. So, the algorithm not only speeds up the mining process but also makes the results more accurate, which means that large datasets can give researchers and practitioners more useful information. This enhancement aids strategic planning and decision-making by diminishing processing time and computational burden while producing more frequent itemsets. The TDA algorithm works better than FP-Growth, which means it can analyze data faster and handle bigger datasets without losing quality. This feature lets businesses make quick, smart choices that lead to more efficiency and new ideas in many fields. The algorithm's streamlined approach also makes it easier to work with large datasets, and its ability to grow makes it even better for frequent itemset mining tasks. The next sections will give a full explanation of the proposed algorithm, including its benefits and uses:-

Input
- Transaction database (DB)
- Minimum support threshold (minsup)

Output
- Identified frequent (recurring) itemsets

Step 1: Database Scanning and Frequent Item Preparation
After each transaction is processed:
1. Scan the entire database.
2. Identify:
   - The set F of all items.
   - The supporters (transactions) for each item in F.
   - The frequent items (those meeting minsup).
3. Sort F in descending order of frequency to form OFIL (Ordered Frequent Item List).
4. Remove all infrequent items.

This step ensures that only the most relevant, high-support items remain, improving both computational efficiency and the overall accuracy of later itemset mining processes.

Step 2: Construction of the TDA

For every transaction row that corresponds to the OFIL structure:
- Insert each frequently occurring item into the appropriate column, following the sorted order in OFIL.

2. The resulting matrix:
   - Highlights usage/consumption patterns.
   - Provides a clear view of item availability and inventory status.
   - Helps determine which items may need restocking.

Maintaining OFIM ensures a clean, organized data structure for pattern mining.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

Step 3: Frequent Itemset Generation
Let c be the number of columns in the TDA.

➢ *Initialize*
- Set c = M, where M is the total number of columns in TDA.

➢ *Process Columns in Reverse Order*

For each column from c = M down to 1:

Case A: When c = 1
1. Compare frequent items in the current column with those in previous columns.
2. Compile their supporters.
3. Represent results as [r, f : n | OFIL], where:
o r = parent frequent item from earlier columns.
o f = frequent item in the current column.
o n = support count.
4. Retrieve the rows of item f from the supporters of column 1.
5. Retrieve the corresponding rows of item r.
6. Remove these extracted rows from TDA to eliminate duplicates and maintain accuracy.

This step clarifies the relationships between items and helps identify patterns in early-column frequent items.

Case B: When c > 1
1. Move to column c and compare its frequent items with those in earlier columns.
2. Compile supporters for each frequent item.
3. Represent results again as [r, f : n | OFIL].
4. Extract rows linked to the repeating parent items.
5. Process the repeating item f according to its order.
6. Remove the extracted rows from TDA.
7. Verify that the remaining TDA structure is consistent and matches the master file.

This makes sure that the generation of frequent itemsets happens in an orderly way and that the relationships between items are correctly represented.

Table 4 shows a complete list of all the frequent item sets that were found in the data analysis. The sets are arranged by their support values, with the ones with the highest frequency at the top. This systematic arrangement makes it easier to find patterns and trends in the dataset. Researchers can find important links between products that might affect how people buy things by looking closely at these common item groups. These kinds of insights are very helpful for creating targeted marketing plans and improving inventory management to better match what customers want. So, looking at frequent item sets not only helps us understand how consumers behave, but it also gives us useful information that we can use to make strategic business decisions.

Table 4. Displays the Created Frequent Item Sets

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

VI. RESULTS AND DISCUSSIONS

The UCI Machine Learning Repository is an important resource for the data mining and knowledge discovery communities. It has a wide range of benchmark and real-world datasets that are necessary for testing the effectiveness of new methods [28]. Researchers can rigorously test their algorithms across a wide range of fields, such as the social sciences, biology, and economics, by using these datasets. This variety not only makes the algorithms stronger, but it also encourages new uses in many different areas. Researchers use these databases to find patterns, back up their results, and learn more about the things they are studying. This study compares the suggested method to the well-known FP Growth algorithm by looking at how many frequent itemsets it finds and how long it takes to get these itemsets from the datasets. The results show that the suggested method makes data mining much more efficient by finding more frequent itemsets and cutting down on computation time by a large amount. This enhancement facilitates the application of these methodologies to larger and more intricate datasets, potentially transforming the extraction of data-driven insights. The experiments were done on a laptop with a 64-bit Windows 10 operating system, Python, 32GB of RAM, and an Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz. Table 6 shows a detailed statistical analysis of the datasets used in this comparative study. These datasets range in size and complexity from small to large. These differences make it easier to fully evaluate the analytical techniques, showing how useful and relevant they are in many different situations.

Table 5: Characteristics of the Test Datasets

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

In algorithmic performance evaluation, employing varied datasets is essential for evaluating the effectiveness of various computational methodologies. This study utilized two separate datasets to methodically evaluate the performance of algorithms under diverse conditions. The results showed a big difference: one algorithm was much faster than the other, while the other was much more accurate. This difference shows how important it is to consider the situation when choosing the best algorithmic strategy for a job. A detailed comparison of the FP-Growth algorithm and the suggested algorithm reveals their strengths and weaknesses, which helps us understand these results better. This analysis ultimately emphasizes the significance of considering both speed and accuracy, among other factors, when selecting an algorithm for practical application.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

➢ *Experiment One*

The experiment used the Poker Hand dataset, which has records of hands of five playing cards taken from a standard deck of 52 cards. There are ten predictive features for analysis because each card has two properties: suit and rank. This dataset has a lot of information that makes it easy to look at different poker hands and their chances of winning. Using machine learning, it is possible to make models that can guess how strong a hand is or find the best ways to play. We did a lot of tests with different minimum support (minsup) values to see how well the proposed approach worked compared to the original FP-Growth algorithm. The Poker Hand dataset offered a practical and realistic environment for evaluating the algorithm's efficacy. The results showed that the machine learning models not only made predictions more accurate, but they also gave us a better understanding of how to play complex games. This progress opens up new possibilities for looking into how AI can make it easier to make decisions in poker and other games that require strategy. Researchers showed that the proposed method worked better than the original FP-Growth algorithm in a number of different situations by changing the minsup parameters. Table 6 shows the results, including how long it took to find the frequency of itemsets and the number of frequent itemsets at different minsup values, like 30%, 45%, 50%, and 60%. These results show how AI could help improve strategic gameplay and decision-making.

Table 6: 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	Apriori	New algorithm
1	30%	84.745	77.288	7.932	1061	864	266
2	45%	82.815	75.267	7.544	837	602	245
3	50%	81.897	74.875	7.165	831	535	94
4	60%	79.197	72.586	7.154	827	411	53

The minimum support (minsup) threshold has a big effect on how well data mining algorithms work. It changes both the number of frequent itemsets created and the time it takes to run. When the minsup values go up, the number of frequent itemsets and the execution times of both the proposed method and the FP-Growth algorithm go down. The proposed method, on the other hand, always runs faster than the FP-Growth algorithm, no matter what minsup threshold is used. This shows that the suggested method is more scalable and efficient, especially when working with large datasets where the cost of computing is very important. The faster execution speed not only makes everything work better, but it also makes it easier to get insights faster when analyzing data. Figure 3 shows how three algorithms compare in terms of performance at four different minsup thresholds. It shows that the proposed method is more efficient than the FP-Growth algorithm. The results show a big increase in execution speed, which shows that the proposed method can handle large amounts of data well. This progress also makes data mining easier and opens the door for more research into how to improve algorithmic performance across a range of applications. The FP-Growth method, on the other hand, needs a lot of memory and time to build multiple conditional sub-trees before it can find frequent itemsets. This can make it less efficient in large data environments.

Table 7: 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	Apriori	New Algorithm
1	10%	3.146	0.686	0.501	123	102	88
2	20%	2.015	0.641	0.488	102	93	42
3	30%	1.722	0.614	0.321	96	83	31
4	50%	1.430	0.561	0.315	81	61	17

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

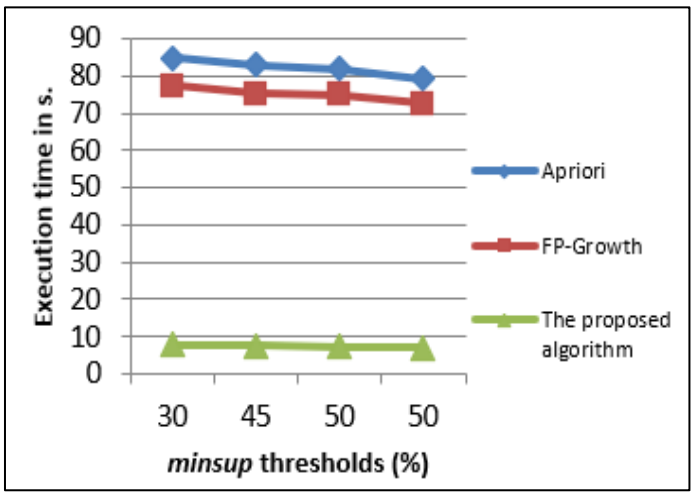

Fig 2: Comparing the Results of the Execution time and the *minsup* Thresholds for the Poker Hand Dataset.

➢ *Experiment two*

The study used the Sepsis Survival Minimal Clinical Records dataset, which is a complete set of 110,204 hospital admissions in Norway between 2011 and 2012. These admissions involved 84,811 patients who had infections, septic shock, sepsis caused by pathogenic microorganisms, or systemic inflammatory response syndrome. The wealth of information in this dataset makes it possible to do a detailed analysis of patient outcomes, which makes it easier to assess the effectiveness of different treatment plans. Researchers are looking for patterns in demographics, clinical interventions, and survival rates that could help improve how sepsis is treated and how patients are cared for in the future. The main prediction task is to use the patient's medical records to figure out if they will live for about nine days after being admitted. Table 7 shows the execution time, the number of frequent item sets found using the FP-Growth algorithm, and the best method for each of the four minimal criteria thresholds: 10%, 20%, 30%, and 50%. This analysis not only shows how the dataset could help doctors make better decisions, but it also shows how carefully the treatment outcomes were measured.

Fig 4 shows that the four different minimum support (minsup) thresholds have very different algorithm execution times, which shows that the performance is very different. The data shows that the chosen minsup threshold has a big effect on execution times, which shows how important it is to choose the right threshold for algorithm efficiency. Algorithm A consistently outperformed the other algorithms tested, even when the minsup levels were lower. This means that Algorithm A is especially good at handling large datasets, making it a strong choice for applications that need a lot of data. The results show that the algorithm can improve data processing and optimize computer resources, making it the best choice for jobs that need to be done quickly and on a large scale.

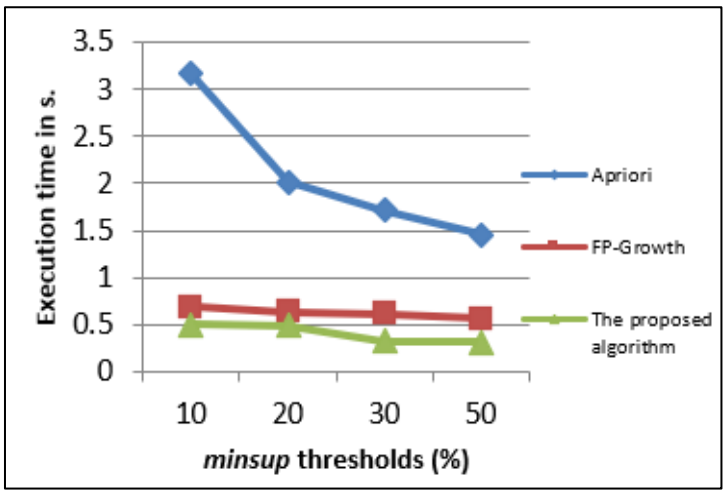

Fig3: Comparing the Results of the Execution time and the *minsup* Thresholds for the Sepsis Survival Minimal Clinical Records Dataset.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

VII. CONCLUSION

In this study, we conducted a comparative analysis of three algorithms—Apriori, FP-Growth, and TDA—focusing on their efficacy in mining frequent itemsets. People like the Apriori algorithm because it's easy to use and can find the best combinations of rules. FP-Growth, on the other hand, is slower because it has to make a lot of conditional sub-trees, which takes a long time to scan the dataset and uses a lot of memory. Our research introduces an enhanced FP-Growth methodology that employs Ordered Frequent Item Lists (OFILs) to generate the TDA, thereby increasing mining efficiency in large data environments.

The evaluation, which looked at how long it took to run with different minimum support (minsup) values, shows that the new method works better than Apriori and FP-Growth. The results show that the program runs much faster when the minsup values are higher. This proves that the suggested method is effective at speeding up data mining and helping people make decisions more quickly. By eliminating the need for conditional sub-tree construction, our method speeds things up and uses less memory. It handles computational resources and efficiency better than the old Apriori and FP-Growth algorithms.

The study acknowledges limitations despite these enhancements, including potential performance variability across different datasets and the need for further optimization in specific contexts. Future research should examine the integration of the proposed methodology with other data mining techniques and assess its applicability across diverse domains. Also, looking into adaptive strategies that can change minsup values on the fly could make the algorithm work even better. This study generally contributes valuable insights to the development of more efficient data mining algorithms that require fewer resources.

The suggested method is a big step forward in making frequent item sets because it gets rid of the need to build conditional sub-trees, which makes the process go faster. This simpler method makes things work better, which is especially helpful when looking at big datasets where old methods often can't keep up with performance levels. This method is better than the FP-Growth method because it uses less memory and works faster. The proposed algorithm's performance benefits become even clearer as the minimum support (minsup) values go up. This shows that it is better than both the FP-Growth and Apriori algorithms. This shows that the method could be a good way to deal with hard data mining jobs.

REFERENCES

[1]. Riadi, I., Herman, H., Fitriah, F., Suprihatin, S., Muis, A., & Yunus, M. 2023. Implementation of association rule using apriori algorithm and frequent pattern growth for inventory control. Jurnal Infotel, 15(4),pp. 369-378.

[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.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec579

[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]. Sohrabi, M. K., & HASANNEJAD, M. H. 2016. Association rule mining using new FP-linked list algorithm.
[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]. Al-Maolegi, M., & Arkok, B. 2014. An improved Apriori algorithm for association rules. arXiv preprint arXiv:1403.3948.
[23]. Yuan, X. 2017. An improved Apriori algorithm for mining association rules. In AIP conference proceedings (Vol. 1820, No. 1). AIP Publishing.
[24]. Han J, Pei J, Yin Y. 2000. Mining frequent patterns without candidate generation (Acm Sigmod Record, 29(2)), pp.1-12.
[25]. 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.
[26]. 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.
[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]. 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.