ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

A More Effective FP-Growth Algorithm for Big Data Using the FP_TDA Algorithm

Abdulkader Mohammed Abdulla Al-Badani[1*]; Abdualmajed Ahmed Ghaleb Al-Khulaid[2]; Abeer A. Shujaaddeen[3]

[1]Faculty of Science and Engineering, Department of Computers, Aljazeera University, IBB, Yemen.
[2]Faculty of Computer Science & Information Systems, Sana'a University, Sana'a, Yemen.
[3]Faculty of Computer Science & Information Systems, Sana'a University, Sana'a, Yemen.

Corresponding Author: Abdulkader Mohammed Abdulla Al-Badani[1*]

Publication Date: 2025/12/01

Abstract: The goal of association rule mining is to identify patterns in big data sets. Businesses may make well-informed decisions based on consumer behavior and preferences by using these links to uncover patterns or correlations that might not be immediately apparent. Apriori and FP-Growth are two examples of algorithms that companies may use to effectively extract insightful information from their data.The association rule method does, however, have certain limitations, 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 efficient algorithm Fp-TDA, based on the FP-Growth algorithm, would reduce the number of frequently formed items and the amount of time spent mining by using the proposed matrix TDA instead of the tree used in those methods. This would result in a significant reduction of the amount of decision-making in large datasets. By reducing redundancy, this method not only speeds up data processing but also increases the correctness of the output. As a result, the Fp-TDA algorithm has the potential to greatly enhance data mining applications, particularly in domains like market research and fraud detection where accuracy and speed are crucial.

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

How to Cite: Abdulkader Mohammed Abdulla Al-Badani; Abdualmajed Ahmed Ghaleb Al-Khulaid; Abeer A. Shujaaddeen (2025) A More Effective FP-Growth Algorithm for Big Data Using the FP_TDA Algorithm.
International Journal of Innovative Science and Research Technology, 10(11), 2109-2119.
https://doi.org/10.38124/ijisrt/25nov1256

I. INTRODUCTION

Manuscripts must be in English. These guidelines Data mining has garnered significant interest from scholars and database practitioners in recent times because to its applications in a variety of domains, including financial forecasting, market strategy, and decision support [1]. Data mining's capacity to reveal hidden patterns and insights from enormous volumes of data, eventually leading to well-informed judgments, is primarily responsible for this spike in interest. More advancements in the industry are anticipated as a result of the growing demand for sophisticated data mining algorithms and processes as businesses continue to see its benefits.

The popular FP-Growth algorithm for association rules is another [2]. The association analysis time of the approach has been much decreased by the FP growth algorithm in comparison to the original association rule algorithm. To solve the problem of multiple scans, it uses a tree structure rather than continually scanning the database to generate itemsets[3]. FP-tree is a tree structure that enables effective data compression and makes it easier to find frequently occurring itemsets without requiring several database queries. Because it offers quicker performance without compromising the integrity of the association rules developed, FP-Growth is especially beneficial for huge datasets.

In this research, the implementation of a standard data mining technique, Pattern-Growth (FP-Growth), will be explored. The association strategy in data mining includes this algorithm. The FP-Growth approach greatly lowers the computational cost by mining frequent itemsets without the need for candidate creation. In order to efficiently find hidden patterns in big datasets, the technique makes use of a compact data structure called the FP-tree.To identify the most common group of data in a set of data, one alternative approach that may be utilized is the FP-Growth itself. For applications where rapid and resource-efficient data processing is essential, this makes it the perfect option. Furthermore, its usefulness in practical situations is further increased by its capacity to manage massive data quantities with less memory

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

utilization.The FP-Tree is a data structure that uses the FP Growth algorithm. Frequent itemsets may be easily extracted from FP-Tree using the FP-Growth method [4].

In data mining, a technique called the apriori algorithm is used to find patterns in a collection of transaction data that often occur (frequent itemsets). This method requires that the dataset be broken up into smaller subgroups. The frequently occurring elements in each subgroup are then looked for. The associations between various things in the data may then be better understood by using the association rules that can be created from these frequently occurring itemsets. Businesses may learn a lot about consumer behavior and make well-informed decisions based on purchase trends by examining these guidelines.Following their discovery, the itemsets will be combined to form larger itemsets, and their frequency will be modified. This method is repeated iteratively until no more frequently occurring itemsets are found. In addition to improving understanding of item linkages, this iterative method optimizes marketing tactics and inventory management. In the end, companies are able to better customize their products to satisfy consumer needs by analyzing common itemsets and the associated association rules.Because they enable us to recognize recurring patterns in consumer purchase behavior and create more effective marketing efforts, apriori algorithms are useful in data analysis and marketing [5].

The frequent pattern growth (FP-Growth) method is an adaption of the Apriori methodology [6]. By building a tree, or FP-Tree, the FP Growth approach finds common item sets [7]. The FP-Tree idea makes FP-Growth a more efficient process.The recently developed and very effective tree-based technique for mining frequently recurring item sets is called FP-Growth [8]. 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 [9] Furthermore, the large database structure cannot be supported by the main memory due to the potentially vast size of the resulting tree [10]. Therefore, the proposed approach employs a novel, two-dimensional array structure based on the FP-Growth algorithm termed the FP-TDA instead of the tree used in earlier techniques. By making it easier to store and retrieve frequently occurring itemsets, the matrix TDA allows for faster computation and result extraction than the traditional tree-based technique.

The rest of the paper is organized as follows: Section 2 presents pertinent work. The Research Method is described in Section 3. The FP-TDA is covered in detail 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.

II. RELATED WORK

There are several FIM-related algorithms in [11,12,13,14]. [15] presents a novel approach to mining association rules with FP-Linked lists. Based on the FP-Growth concept, it has introduced a novel frequent pattern mining technique that uses a bit matrix to extract frequent patterns. This method increases the effectiveness of pattern detection by reducing the search space through the use of a bit matrix representation. As a result, it greatly advances frequent pattern mining by facilitating faster processing and improved scalability when dealing with large datasets.

Assessment of Apriori and FP-Growth Algorithms' Performance (Association Rule Mining) [16]. Using Weka, they assessed two algorithms (Apriori and FP-growth) while accounting for the database scan results (number of occurrences, confidence, and support levels). The findings showed that, especially when dealing with bigger datasets, FP-Growth performed better than Apriori in terms of execution time and memory use. This effectiveness may be ascribed to FP-Growth's capacity to lower the necessary number of database scans, improving association rule mining operations' overall performance.The FP-Growth algorithm clearly performs better than the apriori method.

The efficiency and execution time of the FP-Growth algorithm were superior to those of the Apriori approach when database scan characteristics such as the number of instances, confidence, and support levels were taken into account. FP-Growth is the preferred choice for practitioners dealing with large data since this benefit becomes increasingly apparent as the dataset size grows. Faster processing and easier manipulation of frequent itemsets are made possible by its compact data structure, the FP-tree, which further validates its position as a powerful tool in the data mining sector.This implies that for association rule mining jobs, FP-Growth would be a more suitable option than Apriori.

A more advanced version of the FP-Growth technique for mining description-oriented rules is introduced in [17].They have proposed a unique modification for the definition of gene groups based on the Gene Ontology (GO) FP-Growth algorithm. The results show that the new method allows for quicker rule creation. A new method for using FP-Linked lists to mine association rules is shown in [18]. Based on the FP-Growth idea, it has released a ground-breaking frequent pattern mining approach that uses a linked list structure and a bit matrix to extract frequent patterns.[18] proposes a way to improve the efficiency of mining association rules by combining bit matrices with linked lists. The outcomes demonstrate that our approach outperforms traditional FP-Growth techniques in terms of speed and scalability.

In [19], the Fp-growth algorithm and the Apriori algorithm are used to mine and evaluate traffic accident characteristics. Traffic accident causes and severity are related, as shown by association rule mining. The Apriori method is less computationally efficient for mining frequent item sets by creating and pruning candidate item sets, while having a high accuracy rate. In contrast, the Fp-growth method builds a Fp-tree rather than running several database scans, which improves efficiency and saves memory. Because of this, the Fp-growth method is especially beneficial for big datasets when time and computing resources are crucial.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

The conventional frequent pattern mining techniques that include candidate set generation and testing (Apriori algorithm) and the approach without candidate set generation (FP growth algorithm) are compared in [20]. Every subset of a frequent itemset discovered by the apriori technique must also be frequent. A list of possible things is generated by the apriori algorithm, which then assesses their frequency. By using pattern fragment growth, the FP growth approach searches large databases for recurring patterns. An improved prefix tree structure is used to contain important and concise information about frequent patterns. FP growth detects frequently recurring item sets without creating candidate item sets.

To improve the efficiency of FP-growth by eliminating the necessity for the repeated generation of conditional subtrees, the study in [21] introduces a modified FP-growth (MFP-growth) algorithm. The suggested approach lowers the complexity of the entire frequent pattern tree by using a header table arrangement. To evaluate the operating efficiency of the suggested MFP-growth algorithm with the most advanced machine learning algorithms in terms of runtime, memory consumption, and the efficacy of created rules, four experimental series are carried out using various benchmark datasets. The experimental findings support the MFP-growth algorithm's superiority and highlight its possible applications in diverse settings.

A Signature-Based Tree for Determining Typical Itemsets in [22].The authors of this study suggest a unique tree-based structure that gives transactions precedence over itemsets. By doing this, the issue of support values having a detrimental effect on the generated tree is avoided. Numerous methods have been proposed for frequent item sets mining based on the fp-growth algorithm to accomplish this aim while preserving value, efficacy, and privacy [23]. These techniques not only improve the mining process' efficacy but also deal with important data confidentiality concerns. Using cutting-edge methods, researchers want to develop a more resilient framework that can adjust to different data distributions while maintaining excellent performance across a range of applications.

The study's main contribution is a novel algorithm that efficiently uses the FP-TDA structure to tackle challenging optimization problems in a fraction of the time required by earlier methods. The performance of algorithms that use FP-trees is then improved by this method. Tested on several datasets, our method has continuously outperformed earlier algorithms in terms of speed and accuracy. This novel approach has the potential to revolutionize the optimization industry since it can tackle complex issues more rapidly and precisely. The application of this method to optimization problems beyond those that were studied in this study may potentially be the subject of future research. The use of FP-tree-based algorithms in a variety of domains, including network architecture, finance, and logistics, may yield new advantages and insights if its use is expanded. This growth may result in novel approaches that tackle urgent problems in a range of sectors and further algorithmic theory at the same time.

III. RESEARCH METHOD

An analytical step in the database knowledge discovery process, data mining—also referred to as knowledge discovery in the database, or KDD for short—identifies bias information in the form of hitherto undiscovered patterns in the data or connections between trustworthy data. These trends can yield valuable information that improves strategy planning and aids in decision-making for companies. By recognizing these connections, businesses may spot trends, optimize procedures, and ultimately gain a competitive advantage.By integrating techniques from several computer science fields, such as artificial intelligence, machine learning, statistics, and database systems, data mining is the process of finding new patterns in extremely large data sets. By employing data mining to transform raw data into useful information, organizations may predict future occurrences and tailor their services to meet client expectations. As technology develops, data mining techniques get more sophisticated, making it simpler to extract valuable information from big, complex data sets. It entails sifting through vast volumes of data to find patterns and correlations that might be used to guide decision-making. In today's data-driven world, data mining is an essential tool for companies trying to get insights and maintain their competitiveness [24][25].

A. Association Rule Mining

Association analysis is a data mining technique that may be used to find interesting relationships between a collection of hidden components in a database. One means of expressing this relationship is through association standards [26]. These norms have the power to uncover patterns that show the relationships between various objects or variables, frequently producing revelations that help guide decision-making. Businesses may improve inventory management or improve marketing efforts by utilizing these associations, which are based on consumer behavior and preferences. The purpose of association analysis is to determine how two or more qualities are related. The format of association rules is frequently IF antecedent THEN consequent. Two measures that may be used to assess the strength of an association rule are support and confidence. The percentage of these components combined in the database is known as the confidence value (certainty value) and support value (support value). In particular, the close connections between objects according to the associative laws.

- *Apriori Algorithm:*

In-depth steps are used in this method to find subsets that at least some of the item sets share. Using these common subsets to identify patterns and connections in the data may help with analysis and decision-making. By focusing on these similarities, the approach enhances the overall understanding of the underlying structure of the dataset. Using confidence and support metrics, frequent pattern mining produced the desired outcomes in a range of areas. In domains such as market basket analysis, where an understanding of consumer purchasing trends may lead to more targeted marketing strategies, these products have shown to be quite valuable. Additionally, the information gathered from routine pattern mining will improve product recommendations and inventory

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

management, ultimately increasing customer satisfaction and business profitability.

➢ *FP-Growth Algorithm:*

This structure enables efficient traversal and query operations, giving rapid access to the most prevalent patterns in the dataset. Performance while mining for association rules is much enhanced by this method.The two primary phases of the FP-Growth algorithm are mining the frequent itemsets using the FP-tree recursion and constructing the FP-tree.The first column lists the feature item names in decreasing order of support, while the second column has a chain table linking the nodes of the same items in the FP-tree. This structure makes it possible to efficiently traverse the tree in order to calculate itemset frequencies, hence removing the need to generate candidate itemsets directly. The reason FP-Growth is a popular choice for data mining applications with large datasets is that it eliminates the requirement for costly database scans. The two primary phases of the FP-Growth algorithm are FP-tree construction and FP-tree recursion mining for frequent itemsets. The first stage involves building the FP-tree from the transaction database, and the second step involves extracting the frequently occurring itemsets. Using the compact transaction representation of the FP-tree, the approach efficiently recognizes patterns by detecting high-frequency item pairings. During the building phase, the approach constantly scans the dataset and adds transactions to the tree to generate the FP-tree. After the FP-tree is built, the mining step iteratively builds conditional FP-trees by removing frequently occurring itemsets from the tree.

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

Dataset D as input; support threshold min_sup

➢ *FP-Tree as the Output*

- The frequent 1 item set L1 may be obtained by first navigating through the dataset, calculating the support of each feature item, sorting in decreasing order, and then filtering out the infrequent items using min_sup. One way to acquire the frequent 1 item set L1 is to sort the dataset in decreasing order, calculate the support of each feature item, and then use min_sup to filter away the infrequent items. Following the acquisition of the frequent 1 item set L1, the frequent items from L1 are aggregated to produce the candidate 2 item sets (C2). To locate the often occurring two-item set L2, you will next need to filter the dataset using the same minimal support threshold after determining the support for each candidate in C2.
- Construct the FP-tree's root node, give it the value T, and set its content to "null". Set the connection to null and make a table of frequently used objects. To go on to the second iteration of the dataset, follow these steps. Update the counts of frequently used items for every transaction by moving through the tree from the root node T to begin the second iteration of the dataset. If there are any items missing from the tree, create a new node for them and link them correctly to maintain the hierarchical structure of the FP-tree.
- For the D do transaction.
- The frequently occurring items will be filtered based on L1, the items in the transactions will be sorted based on the feature item order in L1, and they will be recorded as P. P will then be analyzed to identify any trends or patterns that might inform future decisions about inventory management. This process will ensure that popular products are readily available, increasing customer satisfaction and speeding up processes.
- P should be inserted into T. If T contains a P prefix, then we should raise the count of each prefix node by 1. There should be a new node with a count of 1 for the item that comes after the prefix if T does not have a P prefix. This technique ensures that each prefix and its associated counts are correctly recorded in the data structure, enabling efficient item manipulation and retrieval.
- In the table of frequently used items, modify the relevant links. Make sure the relevant links in the table of frequently used things are updated to point readers to the most recent resources. A better user experience overall and better navigation will make it easier for people to access the information they need fast.

The FP-tree structure is integral to the efficiency of the FP-Growth algorithm in mining frequent itemsets. This structure includes a header table, where individual items are listed alongside their frequencies in descending order. As illustrated in Figure 1, which depicts the FP-tree generated from the transactional dataset in Table 1, this approach eliminates the need for creating candidate sets. The FP-tree's ability to traverse the dataset in a compressed manner significantly enhances the retrieval speed of itemsets that meet the minimum support criterion. Each node within the FP-tree represents an item and its corresponding frequency count, facilitating the efficient mining of frequently occurring itemsets. Consequently, the FP-tree structure not only optimizes the mining process but also ensures that the algorithm operates with high efficiency by avoiding the computational overhead associated with candidate set generation.

Table 1 A Dataset with Nine Transactions.

TID	List of items
T1	P1,P2,P5
T2	P2,P4
T3	P2,P3
T4	P1,P2,P4
T5	P1,P3
T6	P2,P3

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

T7	P1,P3
T8	P1,P2,P3,P5
T9	P1,P2,P3

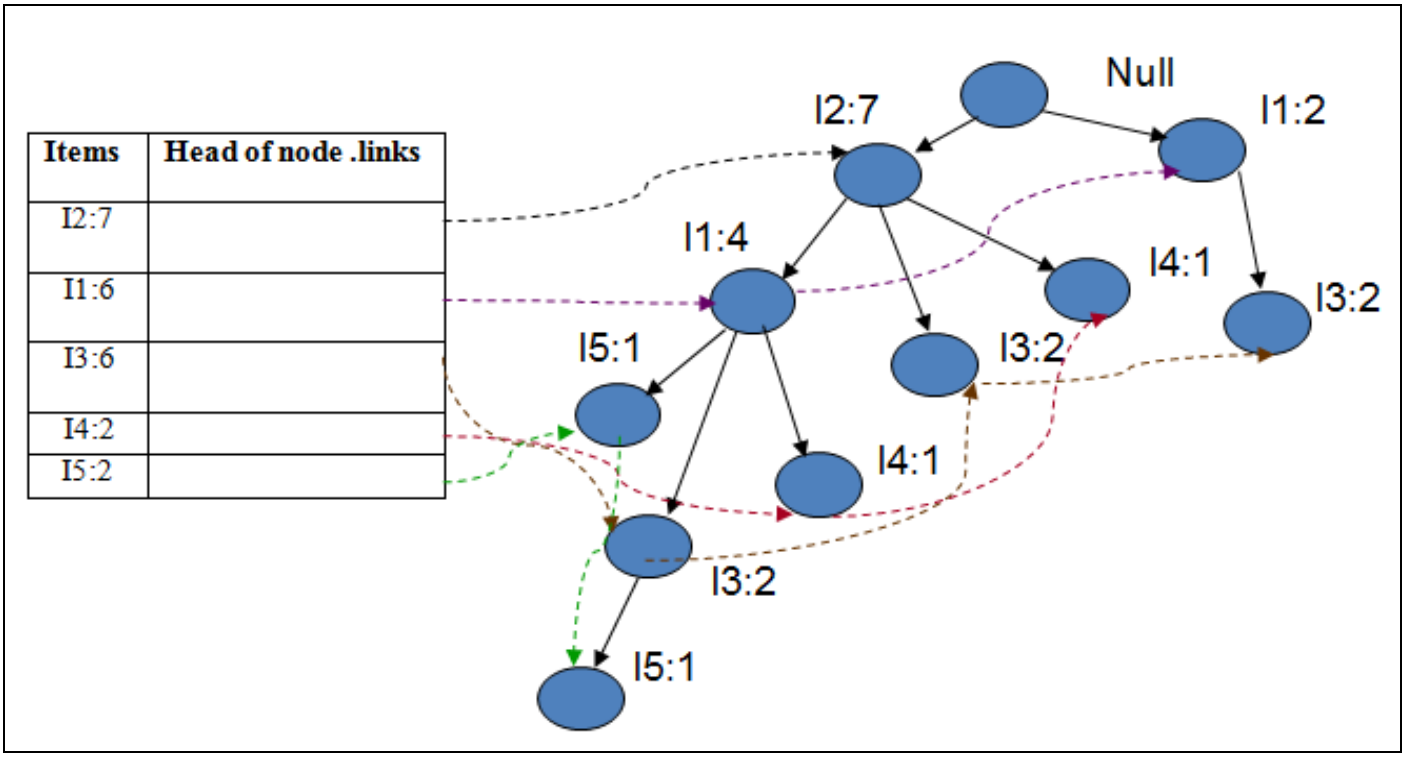

Fig 1 An Example of FP-Tree(Minsup=50%).

Table 2 displays the frequent itemsets that were generated.

Table 2 The Discovered Frequent Item Sets by FP-Growth Algorithm.

TID	Conditional FP-tree	Frequent Item Sets
P5	<P2:2,P1:2>	{P2,P5:2}, {P1,P5:2}, {P2,P1,P5:2}
P4	<I2:2>	{P2,P4:2}
P3	<P2:4,P1:2>,<P1:2>	{P2,P3:4},{P1,P3:4}, {P2,P1,P3:2}
P1	<P2:4>	{P2,P1:4}

Identify applicable funding agency here. If none, delete this text box.

IV. TWO-DIMENSIONAL ARRAY

A two-dimensional array (TDA) is employed to effectively summarize transactional databases by including all frequently recurring itemsets. These itemsets are organized in descending order of their support values. The TDA is structured as an N×M matrix, where M represents the maximum number of regularly ordered products and N denotes the total number of transactions. Each cell in the array contains the support value for the corresponding itemset, enabling swift and efficient extraction of critical data regarding frequently used itemsets from the database. This structured approach to data organization facilitates rapid information retrieval and analysis, significantly enhancing decision-making processes in marketing and inventory management. By concentrating on the most commonly purchased items, businesses can better tailor their offerings to align with customer preferences, thereby optimizing their marketing strategies and inventory control.

The proposed method generates a list of frequently occurring items from transactional data, organized in descending order of occurrence. This hierarchical arrangement is crucial as it influences the construction of the Transactional Data Analysis (TDA). Specifically, the list includes potential itemsets for each transaction that meet or exceed the minimum support (minsup) threshold. These itemsets are further categorized into Ordered Frequent Itemsets Lists (OFILs), which systematically compile frequently occurring itemsets. This structured approach ensures that the analysis is both comprehensive and efficient, facilitating a more accurate identification of significant patterns within the data. Thus, the ordered listing not only enhances the clarity of the data but also optimizes the analytical process, underscoring its importance in transactional data analysis. In order to create the TDA, the OFILs are used to concatenate the frequent itemsets while preserving the frequency decreasing order. Throughout the

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

construction process, this strategy ensures that the most critical itemsets are given priority, increasing the TDA's efficacy and efficiency. Focusing on itemsets with higher frequency may help the model better capture the underlying patterns and linkages in the transaction data.By prioritizing the most significant items, this method minimizes the search space and facilitates effective mining of regularly repeating itemsets. Non-frequent candidates' leftover itemsets are thrown out. In this sequence, the rightmost column in Table 3 lists the items that are often used in each transaction. These frequently occurring itemsets are essential for finding connections and patterns in the data.

Table 3 Transactional Dataset with OFILs.

TID	List of items	OFILs.
T1	P1,P2,P5	P2,P1,P5
T2	P2,P4	P2,P4
T3	P2,P3	P2,P3
T4	P1,P2,P4	P2,P1,P4
T5	P1,P3	P1.P3
T6	P2,P3	P2,P3
T7	P1,P3	P1,P3
T8	P1,P2,P3,P5	P2,P1,P3,P5
T9	P1,P2,P5	P2,P1,P3

The organization of frequently occurring items within transactions is crucial for effective data analysis and retrieval. As demonstrated in Table 3, the transaction P1, P2, I5 is reordered to P2, P1, P5 based on the list of frequently occurring items. To facilitate this process, an empty Transaction Data Analysis (TDA) matrix with dimensions N×M is initialized with "0" values. During matrix generation, each Ordered Frequent Item List (OFIL) is examined sequentially, ensuring that the most frequently occurring items are prioritized. This method enhances the accuracy of representing the significance of each item, thereby providing deeper insights into transaction patterns and linkages. Items are subsequently added to the appropriate rows and columns of the matrix, with each OFIL undergoing this procedure in turn. Table 4 offers a detailed description of the final TDA matrix after processing all OFILs listed in Table 3. This table not only outlines the key characteristics and relationships identified but also highlights any anomalies or trends that warrant further investigation. Such detailed analysis paves the way for more targeted research and potential operational improvements. In summary, the systematic organization and examination of OFILs within the TDA matrix enable a more nuanced understanding of transaction data, facilitating more informed decision-making and strategic planning.

Table 4 The TDA.

T1	P2	P1	P5	0
T2	P2	P4	0	0
T3	P2	P3	0	0
T4	P2	P1	P4	0
T5	P1	P3	0	0
T6	P2	P3	0	0
T7	P1	P3	0	0
T8	P2	P1	P3	P5
T9	P2	P1	P3	0

V. FP-TDA ALGORITHM

In this Section, a new algorithm based on FP-tree and the TDA structure is presented, which is called Fp-Tda.

Because it takes a long time to generate an FP-tree and identify several frequently recurring itemsets, the basic FP-Growth technique is not appropriate for large data sets, even though it may perform well for small ones. It may not be able to fit in the main memory due to the expanding FP-tree. A minsup threshold and the Fp-Tda are among the inputs needed to find regularly repeating itemsets. The One-Itemset-at-a-Time Mining (Fp-Tda) method addresses the challenge of memory limitations by employing a two-dimensional array that is dynamically updated as new itemsets are identified. This approach is particularly advantageous when dealing with large datasets, as it ensures scalability and efficient memory utilization. By concentrating on one itemset at a time, the FP-TDA method can process substantial data volumes without encountering memory constraints. Consequently, this technique not only enhances the capacity to manage extensive datasets but also optimizes overall computational efficiency.

The detailed FP-TDA algorithm is as follows.

A. Procedure FP-TDA

➢ *Procedure FP-TDA Calculate*

- Input: Candidate Itemset C

✓ Output: the support of candidate itemset C

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

- Make the TDA. 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 guarantees that all pertinent information is arranged and readily available, facilitating effective inventory tracking and management. Frequent TDA updates will enhance procurement and stock level decision-making processes and help maintain accuracy.

➢ *Generation of Frequent Item Sets.*

- Assume that the TDA column number is c.
- For (c= M; c>=1; c--)

{
If c=1 Then Do
{

Along with its predecessors, the current column (c) compares and compiles the group of often occurring items and their proponents. Let r be the parent frequent item of the preceding columns and f be the current frequent item in column (c). The result is [r, f: n | OFIL]. With this output style, the relationships between items that appear often in different columns may be easily observed. Finding the parent frequent item (r) and its matching current frequent item (f) of the dataset allows us to quickly analyze the support patterns and understand the relationships between these items.

The prior columns of item f are compiled from the present column's (c) supporters, and the corresponding rows of item r are taken from the collection of often occurring items. These rows should also be deleted from the TDA. This procedure guarantees that the dataset we use is clean and devoid of any duplicates that may distort our research. By methodically removing these rows, we improve the precision of our findings and expedite the processing stages that follow.

}
Else Do
{

The process of analyzing frequently used items involves a systematic comparison and aggregation of data across multiple columns. Initially, one must proceed to column (c) and compare the group of often used items, ensuring to gather supporters for each item before advancing further. Let r represent the parent frequent item from the preceding columns, while f denotes the current frequent item in column (c). The outcome of this analysis is represented as [r, f: n | OFIL], where 'n' signifies the number of occurrences and 'OFIL' stands for the Ordered Frequent Item List. This methodical approach allows for the identification and tracking of item frequency patterns, thereby facilitating a deeper understanding of the data set's structure and trends.

To identify and manage repeated parent items within the dataset, it is essential to extract the relevant rows. Specifically, the item labeled 'f,' which appears multiple times, must be addressed based on its sequence of occurrence. These identified rows should then be removed from the TDA (Total Data Analysis) to ensure accuracy and prevent redundancy. By systematically extracting and eliminating these repeated entries, the integrity of the dataset is maintained, facilitating more reliable and precise analysis. This methodical approach not only enhances data quality but also streamlines subsequent analytical processes.

}
}

The process of generating frequent itemsets in the recommended algorithm begins with determining the support for each item in the final column of the Transaction Data Array (TDA). The preceding column is utilized to distinguish the items in the current column. Once the support for the various items in the current column is calculated and those that meet the support threshold are identified, the previous column is removed from the current column. Subsequently, the corresponding rows are eliminated from the TDA. This procedure is iteratively repeated for each column until all columns have been processed.

The recommended algorithm for generating frequent itemsets initiates by calculating the support for each item in the final column of the Transaction Data Array (TDA). The penultimate column is employed to differentiate the items in the current column. Upon determining the support values for the items in the current column and identifying those that meet the support threshold, the items from the previous column are removed from the current column. Consequently, the rows corresponding to these items are eliminated from the TDA. This iterative process is repeated for each column, ensuring a systematic reduction of the dataset until all columns have been processed. This methodical approach ensures the efficient identification of frequent itemsets, which is crucial for subsequent data analysis and mining tasks.

The common item sets that were developed are shown in Table 5. The item sets are arranged in the table according to their support values, with the sets that appear most frequently at the top. It is possible to find patterns and trends in the data using this information. These patterns provide valuable information for marketing and inventory management strategies by allowing researchers to identify the item combinations that are often purchased together. Additionally, this data can assist in forecasting future consumer behavior and optimizing product placement.

Table 5 Displays the Created Frequent Item Sets Frequent Item Sets

Frequent Item Sets
{P2,P3:2}, {P2,P1,P5:2}
{P2,P1,P3:2}

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

VI. RESULTS AND DISCUSSIONS

To comprehensively assess the efficacy of the proposed technique, we employed datasets sourced from the UCI Machine Learning Repository. This repository is a renowned and widely recognized resource within the data mining and knowledge discovery in databases (KDD) communities, providing both benchmark and real-world datasets[28]. By leveraging these datasets, we ensured that our evaluation was grounded in a robust and diverse set of data, thereby enhancing the reliability and generalizability of our findings. The selection of datasets from such a reputable source underscores the methodological rigor of our study and aligns with established practices in the field. Consequently, the use of these datasets not only validates the performance of our technique but also facilitates meaningful comparisons with existing methodologies.

The availability of comprehensive datasets serves as an indispensable resource for both academics and practitioners, facilitating the systematic testing and comparison of methodologies against established benchmarks. Utilizing this repository enhances the credibility and applicability of our algorithm by demonstrating its effectiveness across a wide range of scenarios. This process not only validates the robustness of the algorithm but also ensures its relevance and utility in practical applications. Furthermore, the diversity of the datasets allows for an extensive evaluation of the method across various data distributions and contexts. Such a comprehensive assessment not only underscores the algorithm's robustness but also highlights potential areas for further development, thereby paving the way for continued research and advancement in the field.

The experiments were conducted on a laptop equipped with a 64-bit Windows 10 with Python, 32GB of RAM, and Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz 2.71 GHz.. Table 6 presents a comprehensive statistical summary of the datasets utilized in this comparative analysis. These datasets exhibit a wide range of sizes and complexities, from small to large-scale. Key metrics such as mean, median, and standard deviation are provided for each dataset, offering crucial insights into their respective characteristics. This detailed statistical information is essential for understanding the variability and central tendencies within the datasets, thereby facilitating a more nuanced interpretation of the experimental results.

Table 6 Characteristics of the Test Datasets

Datasets	Size	#Transactions
Printed Circuit Board Processed Image	31.8MB	507324
Bitcoin Heist Ransomware Address	224MB	473376

- *Experiment One:*

The dataset employed in Experiment One was the Printed Circuit Board (PCB) Processed Image dataset, which serves as a representative example of big data in the context of manufacturing processes. This dataset comprises modified transactions that simulate real-world scenarios in PCB production, thereby providing a robust foundation for analyzing the efficacy of data processing techniques. By leveraging this dataset, the experiment aims to evaluate the performance of various algorithms in handling large-scale data and extracting meaningful insights. The choice of this dataset is particularly pertinent due to its complexity and the relevance of PCBs in modern electronics, making it an ideal candidate for testing the scalability and accuracy of data processing methodologies. Consequently, the findings from this experiment are expected to contribute significantly to the optimization of data handling strategies in industrial applications, underscoring the importance of efficient data management in enhancing production efficiency.

The analysis of Table 7 reveals significant insights into the performance of the proposed method compared to the original FP-Growth algorithm, specifically in terms of accuracy and efficiency across varying minimum support (minsup) values. The results, categorized by the frequency of itemsets and the execution time required for their identification, indicate that the proposed method consistently outperforms the FP-Growth algorithm at all tested minsup thresholds (0.001%, 0.002%, 0.003%, and 0.004%). This consistent superiority is evident in both the precision of the itemsets identified and the reduced computational time, highlighting the robustness of the proposed approach. Furthermore, the data in Table 7 underscores how different minsup values influence the efficacy of the proposed method, thereby providing a comprehensive understanding of its operational advantages. These findings substantiate the proposed method's potential as a more effective alternative for frequent itemset mining, offering enhanced performance across diverse operational parameters.

Increasing the minimum support (minsup) values in data mining algorithms has a significant impact on both the frequency of commonly encountered itemsets and the execution duration of the algorithms. Specifically, as minsup values rise, there is a notable reduction in the number of frequent itemsets generated and a corresponding decrease in the execution time for both the proposed method and the FP-Growth algorithm. Notably, even with variations in the minsup threshold, the proposed method consistently outperforms the FP-Growth algorithm in terms of speed. This is evidenced by the data presented in Figure 3, which illustrates the execution time performance of the two algorithms across four distinct minsup thresholds. The results clearly indicate that higher minsup values lead to more efficient execution times and fewer itemsets, underscoring the efficiency of the proposed method relative to the FP-Growth algorithm.

The proposed approach significantly enhances processing efficiency while maintaining the integrity of the mining procedure, distinguishing itself from the FP-Growth algorithm. This distinction is particularly pronounced when

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

handling large datasets, where the FP-Growth algorithm often struggles due to slower processing speeds and substantial memory requirements. In contrast, the proposed method facilitates faster data analysis and the extraction of valuable insights with reduced processing overhead. The FP-Growth algorithm's reliance on constructing numerous conditional sub-trees results in a considerable demand for memory and time, leading to the generation of an extensive number of frequent itemsets. By optimizing these aspects, the proposed approach offers a more efficient and scalable solution for data mining tasks involving large datasets.

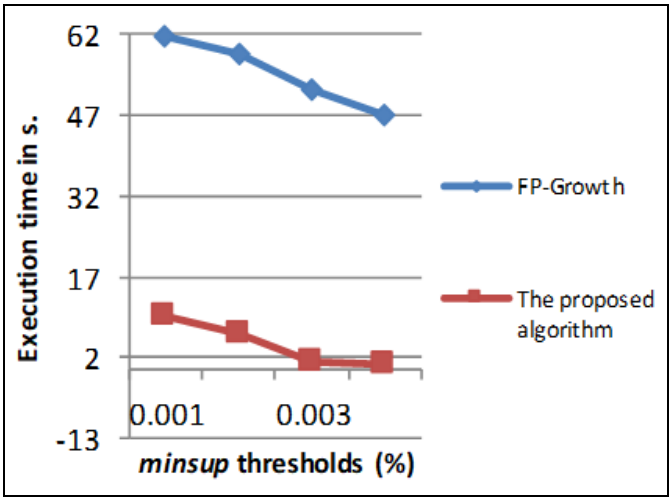

Fig 2 Comparing the Printed Circuit Board Processed Image Dataset's Execution Time and Minsup Threshold Results.

➢ *Experiment Two:*

The dataset utilized in Experiment Two was the Bitcoin Heist Ransomware Address dataset, a comprehensive resource that provides detailed information on Bitcoin addresses associated with ransomware activities. This dataset was selected due to its relevance in analyzing patterns of illicit financial transactions and its utility in advancing cybersecurity research. Specifically, the dataset includes features such as transaction history, address labels, and behavioral patterns, which are critical for identifying and classifying ransomware-related activities. By leveraging this dataset, the experiment aimed to develop a robust machine learning model capable of detecting and predicting ransomware addresses with high accuracy. The inclusion of this dataset not only ensured the reliability of the analysis but also facilitated the exploration of novel approaches to combat ransomware threats. Thus, the use of the Bitcoin Heist Ransomware Address dataset was instrumental in achieving the experiment's objectives and advancing the understanding of ransomware behaviors.

Table 7 Results of a Comparison Using Printed Circuit Board Processed Image Dataset are Shown.

No.	Minsup	Execution time per seconds (s)		# Discovered Frequent Item Sets	
		FP-Growth	New algorithm	FP-Growth	New algorithm
1	0.001%	61.623	9.615	279379	1375
2	0.002%	58.164	6.220	164260	813
3	0.003%	51.839	1.462	98401	348
4	0.004%	46.923	0.791	54695	20

The patterns discerned from the dataset offer valuable insights for researchers aiming to create algorithms capable of reliably predicting the edibility of diverse mushroom species. Table 8 illustrates the execution time, the quantity of frequent item sets identified by the FP-Growth algorithm, and the recommended approach under varying minimum support thresholds (0.003%, 0.004%, 0.005%, and 0.006%). These findings underscore the efficacy of the FP-Growth algorithm in processing large datasets and identifying critical patterns that can inform the development of predictive models. By adjusting the minimum support criteria, researchers can optimize the balance between computational efficiency and the comprehensiveness of the item sets, thereby enhancing the robustness of the predictive algorithms.

Table 8 Results of a Comparison Using Bitcoin Heist Ransomware Address Dataset are Shown.

No.	Minsup	Execution time per seconds (s)		# Discovered Frequent Item Sets	
		FP-Growth	New Algorithm	FP-Growth	New Algorithm
1	0.003%	27.416	3.230	102156	545
2	0.004%	24.181	1.750	66440	253
3	0.005%	22.129	1.060	43114	86
4	0.006%	21.360	0.723	30751	33

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

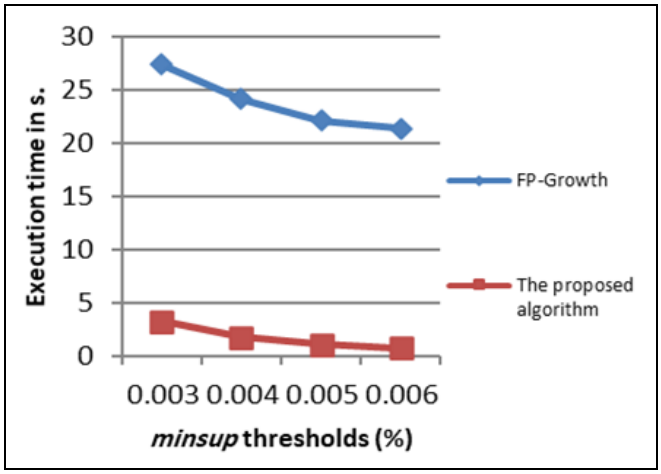

Fig 3 Contrasting the Outcomes of the Bitcoin Heist Ransomware Address Dataset's Minsup Thresholds and Execution Time.

The analysis of the graph reveals a clear inverse relationship between the minimum support (minsup) threshold and both the number of frequently generated itemsets and the execution time of the three algorithms under consideration. As the minsup threshold increases, there is a notable reduction in the number of itemsets that the algorithms must process, which in turn leads to a significant decrease in execution time. This finding underscores the potential for enhancing algorithmic efficiency by strategically adjusting the minsup threshold. By increasing this parameter, practitioners can effectively streamline computational demands while still maintaining a balance with the comprehensiveness of the itemsets identified. Consequently, the ability to fine-tune the minsup threshold offers a valuable tool for optimizing resource allocation and performance in data processing tasks. In the context of large datasets where processing time and memory utilization are critical factors, optimizing these aspects becomes imperative for efficient data handling. One significant improvement is the implementation of a higher minimum support (minsup) threshold in data mining tasks. By increasing the minsup threshold, the algorithm can effectively reduce the number of candidate patterns, thereby accelerating execution times and reducing memory consumption. This approach not only enhances computational efficiency but also yields more manageable and interpretable results by focusing on the most significant patterns. Consequently, adopting a higher minsup threshold is a strategic method to streamline data processing, making it a valuable consideration for researchers and practitioners dealing with extensive datasets.

VII. CONCLUSION

This study proposes an enhanced FP-Growth method to improve the efficiency of mining frequent itemsets in big data environments. The proposed approach leverages Ordered Frequent Item Lists (OFILs) to construct an two-dimensional array (fp_TDA), which significantly optimizes the mining process. By utilizing fp_TDA, the method effectively reduces the number of frequent itemsets generated, thereby streamlining the extraction process and enhancing computational efficiency. Consequently, this technique not only accelerates data processing but also minimizes resource consumption, making it a valuable tool for handling large-scale datasets.

The proposed method effectively enhances system performance by reducing execution time and memory consumption through the targeted deletion of infrequently accessed objects. This deletion is guided by a comprehensive analysis of object usage patterns and their relevance to current system requirements, ensuring that only non-essential objects are removed. As a result, the method significantly optimizes resource utilization and computational efficiency. To evaluate its efficacy, the execution times of the proposed algorithm and the FP-Growth algorithm were compared across various minimum support (minsup) values. The results clearly demonstrate that the proposed algorithm outperforms the FP-Growth algorithm. Unlike the proposed method, the FP-Growth algorithm requires the construction of numerous conditional sub-trees before generating a substantial number of frequent item sets. This process is both time-intensive and memory-demanding, underscoring the superior efficiency of the proposed approach. By addressing these computational bottlenecks, the proposed method offers a more scalable and resource-efficient solution for frequent itemset mining.

The proposed algorithm demonstrates a marked improvement in generating frequent item sets by eliminating the need to construct conditional sub-trees, a process inherent to the FP-Growth algorithm. This innovation significantly reduces both execution time and memory usage, thereby enhancing the algorithm's efficiency and scalability. Empirical results indicate that the performance gains of the proposed algorithm become increasingly evident with higher minsup values, underscoring its superiority over the FP-Growth algorithm. Consequently, this advancement not only optimizes computational resources but also offers a robust solution for large-scale data mining applications.

REFERENCES

[1]. WU, Bo, et al. 2008. An efficient frequent patterns mining algorithm based on apriori algorithm and the FP-tree structure. In: 2008 Third International Conference on Convergence and Hybrid Information Technology. IEEE. pp. 1099-1102.

[2]. 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.

[3]. 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.

[4]. SIAHAAN, Andysah Putera Utama; IKHWAN, Ali; ARYZA, Solly.2018. A novelty of data mining for promoting education based on FP-growth algorithm.

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

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov1256

[6]. 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.

[7]. 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.

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

[9]. 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.

[10]. 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.

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

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

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

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

[15]. Sohrabi, M. K., & HASANNEJAD, M. H. 2016. Association rule mining using new FP-linked list algorithm.

[16]. 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.

[17]. 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.

[18]. 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.

[19]. Bao, Y. Tang, B. Yang, X. Wang, J. Chen, and H. Xiong,2025. "Mining analysis of traffic accident features based on fp-growth algorithm and apriori algorithm," vol. 13486, no. Cvaa 2024, pp. 1–5.

[20]. M. Kavitha and M. S. T. T. Selvi.2016. "Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Pros and Cons," Int. J. Comput. Sci. Trends Technol. (IJCS T), vol. 4, no. 4, pp. 161–164.

[21]. M. Shawkat, M. Badawi, and S. El.2021. "An optimized FP - growth algorithm for discovery of association rules," J. Supercomput., no. 0123456789.

[22]. M. El Hadi Benelhadj, M. M. Deye, and Y. Sliman. 2023. "Signaturebased tree for finding frequent itemsets," Journal of Communications Software and Systems, pp. 70–80.

[23]. S. Bhise and S. Kale. 2017. "Effieient algorithms to find frequent itemset using data mining," Int. Res. J. Eng. Technol., pp. 2645–2648.

[24]. AL-ZAWAIDAH, Farah Hanna; JBARA, Yosef Hasan; MARWAN, A. L. 2011." An Improved Algorithm For Mining Association Rules In Large Databases",World Of Computer Science And Information Technology Journal, 1.7,pp. 311-316.

[25]. Dr. Suyanto, S. M. 2017."Data Mining Untuk Klasifikasi Dan Klasterisasi Data", Bandung: Informatika.

[26]. WINARTI, Titin; INDRIYAWATI, Henny. 2023. Data Mining Modeling Feasibility Patterns of Graduates Ability With Stakeholder Needs Using Apriori Algorithm. International Journal of Information Technology and Business, 4.2,pp. 55-60.

[27]. 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.

[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 Californial, Department of Information and Computer Science.