UNVEILING HIDDEN PATTERNS: EXPLORING THE POTENTIAL OF FREQUENT MINING ALGORITHM

A MAJOR PROJECT-II REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:
SONALI RANI
2K22/ISY/19

Under the Supervision of

Dr. RITU AGARWAL

DEPARTMENT OF INFORMATION AND TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi – 110042

May, 2024

ACKNOWLEDGEMENTS

I am thankful to my supervisor Dr. Ritu Agarwal, Assistant Professor in the Department of Information Technology at Delhi Technological University, Delhi and all the department's faculty members. They all assisted me whenever I needed any help. This work would not be possible without their direction and assistance. I am also grateful to the IT department for providing the various resources required to complete this work.

SONALI RANI
2K22/ISY/19

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I Sonali Rani hereby certify that the work which is being presented in the thesis entitled "Unveiling Hidden Patterns: Exploring the potential of frequent mining algorithm" in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of Information Technology, Delhi Technological University is an authentic record of my own work carried out during the period from 2022 to 2024 under the supervision of Dr. Ritu Agarwal.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other institute.

Place: Delhi **SONALI RANI**

Date: 20/05/2024 **2K22/ISY/19**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that **Sonali Rani** (2K22/ISY/19) has carried out their search work presented in this thesis entitled "**Unveiling Hidden Patterns: Exploring the potential of frequent mining algorithm**" for the award of **Master of Technology** from Department of Information Technology, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution to the best of my knowledge.

Date: 20/05/2024

Dr. RITU AGARWAL

Asst. Professor

Department of Information Technology

DELHI TECHNOLOGICAL UNIVERSITY

ABSTRACT

In the field of data mining, frequent pattern mining has become essential and is attracting a lot of interest from academics. This task's primary objective is to identify repeating subgroups within set sequences. These kinds of projects are important in many different data mining fields, including web mining, association rule discovery, classification, clustering, and market analysis. Many frameworks have been created all through time to make regular pattern mining easier, with the support-based approach being the most well-known.

In order to work, the support-based framework looks for item sets that have a frequency threshold over which they fall. This cutoff is used as a standard to assess how important the patterns are in the dataset. Through the process of identifying frequently occurring item sets, analysts might uncover significant patterns and correlations within the data.

This review paper explores multiple algorithms for frequent mining and provides brief explanations for each of them. Apriori, FP-Growth, and Eclat, three of the most popular techniques in the area, are among the algorithms examined. Every algorithm has a unique collection of benefits, limitations, and guiding ideas that enable it to be applied to many situations and datasets. After performing transaction aggregation on the dataset, the research concludes with a comparative examination of frequently used pattern mining approaches, in addition to examining each of these algorithms separately. This comparison analysis compares the algorithms based on a number of important factors, including memory consumption, computational complexity, scalability, and adaptability to various types of data. By studying these variables, researchers can find out more about the advantages and disadvantages of each strategy, which can help them choose the most effective method for a particular mining.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 General

Making decisions requires the use of data analytics. Such discoveries from pattern analysis have several benefits, such as increased profitability, lower expenses, and a competitive edge. However, mining the hidden patterns of the frequent item sets becomes more time-consuming as the dataset develops. Because it can find the recurring links between many objects in a data set and describe them as association rules, frequent pattern mining (FPM) is among the most important technologies [1].

Figure 1 illustrates a typical data analytics practice's whole deployment process. Prior to pre-processing and transformation, the dataset is first chosen from its origin database and entered into the target data sets. It is then mined for important patterns that may be analyzed or assessed to produce meaningful knowledge. Data mining is one of the stages that is crucial in identifying similar patterns that may be present often in the converted data[4].

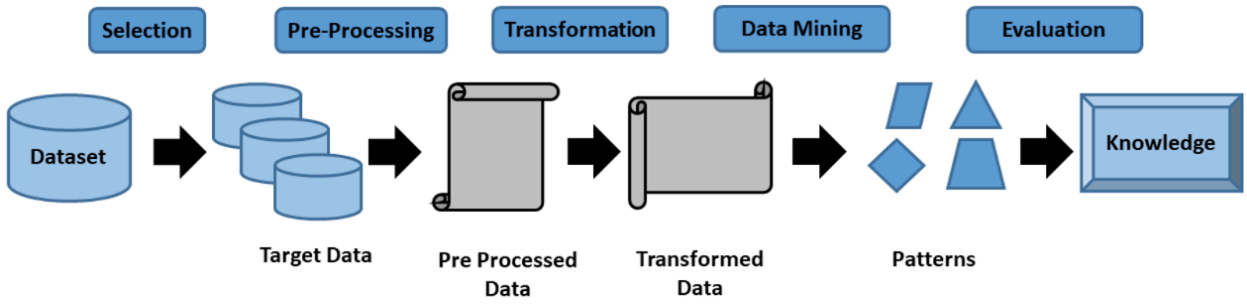


Fig 1.1: Process of mining and knowledge database

FPM concerns have so many applications to different data extracting work including classification, clustering, and outlier analysis, many academics have examined them in great detail [3][7]. FPM is crucial to performing various data mining jobs in order to enhance the process for classifying or clustering a set of data and identifying outliers or anomalies in a data set. In addition, FPM has numerous applications in a variety of fields, including the analysis of

biological and spatiotemporal data and the detection of software bugs [3][4]. Finding the underlying patterns that commonly appear in a data set is a critical step in developing the association rules that will be utilized in data analysis. Different academics have developed a variety of strategies to improve the FPM methodology [2]. The performance of the present FPM methods still has to be improved, however, as the majority of the current algorithms are not suitable for mining a large data set with an increasing amount of data.

The following are the main challenges that the majority of FPM algorithms face:

i. High computational time required

ii. Massive memory usage when the technique is used to find all the hidden frequent patterns.

1.2 Data Mining Tasks and Models

Data mining encompasses several essential tasks, including Classification, Clustering, Regression, and Association rule mining. Among these, Association rule mining stands out as a particularly intriguing area of research.

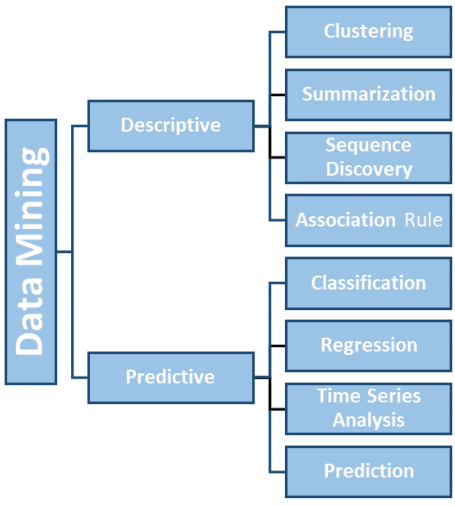


Fig. 1.2: Models and Tasks in Data Mining

Predictive Model

Within the domain of Predictive Models, historical data is utilized to make predictions about specific attribute values. To predict the values of dependent variables, these models rely on independent variables [10]. One of the key responsibilities in data mining is classification, which is grouping data into predefined groupings. Among the popular classification methods are Random Forest and Support Vector Machine.

Descriptive Model

Descriptive models are primarily used to find patterns, correlations, and trends in data in order to provide a thorough description of its properties. The descriptive model is used to carry out the following data mining tasks:

- In order to ensure that data inside one cluster have common qualities while differentiating from data in other clusters, clustering involves grouping data based on similarities.

- In order to extract high-level, concise information from the dataset, summarization involves characterizing and generalizing data.

- By discovering associations between often recurring elements in the data, association rule mining helps businesses find products that are frequently purchased.

- Discovering sequential patterns in data—where interactions are time-based but patterns resemble associations—is the main goal of sequence discovery, sometimes referred to as sequential analysis. For example, those who buy laptops could within a week also buy speakers or pen drives.

1.3 Purpose of Research

The primary objective of this study is to use transaction aggregation techniques to increase frequent pattern mining algorithms performance. In frequent pattern mining which aims to find meaningful relations and patterns in large data, traditional methods heavily suffer from memory

blowups and computational complexities. The goal of this research is to provide an optimized way of finding patterns in data by compressing the data-set by grouping related transactions into aggregated units. The goal is to reduce the complexity and the volume of the data, which in turn should help in reducing the computational overhead, memory efficiency and allows Scalability, an ability that will be beneficial for time-consuming algorithms such as Apriori and Eclat.

This study also kept the quality of patterns from aggregation and discovered only correct and meaningful itemsets which are frequently identified. To show which frequent pattern mining algorithm is a better trade-off between memory use and execution time in aggregated datasets, the performance of different algorithms, such as Apriori, Eclat, and FP-growth, were compared. Their efforts want to make frequent pattern mining algorithms no longer impossible to be used for real-world online massive datasets. They have a strong technique for transaction aggregation to ensure frequent pattern mining and succeed in making real contributions to the field of data mining.

CHAPTER 2

LITERATURE REVIEW

2.1 Frequent Pattern Mining

One of the most important techniques in data mining is frequent pattern mining, which examines large datasets to find patterns and regularities. These patterns often occur as sets of recurring items, series of items or substructures in data. It is a vital step towards uncovering hidden correlations and connections between points of information that makes it useful in various fields.

In 1993, Agrawal et al. published a seminal paper on mining association rules among sets of items in large databases which later introduced the concept of frequent pattern mining [1][2][3] . The main objective behind the development of Apriori algorithm was to identify those sets with many common items in transactional information. The technique builds up larger frequent itemsets incrementally and prunes out ones that do not meet a minimum support criteria using bottom-up approach to finding frequent itemsets.

As researchers got more involved in the field, other complex algorithms were developed to counteract Apriori's limitations such as being inefficient when dealing with very large databases whereby it required multiple database scans and much candidate generation. Based on FP-tree structures (Frequent Pattern Growth), one notable improvement made was by Han et al., 2000 [4]. In order to capture the frequency information of itemsets, FP-Growth employs a compressed representation of the database termed an FP-tree (Frequent Pattern Tree). This method eliminates the need to create candidates and do several dataset scans. This leads to a significant boost in efficiency and scalability [8].

Another significant advance is Zaki's Eclat technique, which mixes bottom-up Lattice Traversal with equivalence Class Clustering using a vertical data structure [12][14]. Instead of describing transactions with horizontal item lists, Eclat uses transaction ID lists for each item, which enables efficient support counting through intersection operations. This approach performs particularly effectively on high dimensional, poorly dispersed datasets[16].

Regular pattern mining offers a wide range of applications. By employing frequent pattern mining, market basket analysis is utilized in retail to identify products that buyers commonly purchase together. Inventory control, product positioning, and marketing strategies are then based on this data. Frequent online page viewing habits may be analyzed to help with web usage mining, which improves user experience and website design. In bioinformatics, frequent pattern mining can identify connections between genetic markers and diseases.

2.2 Related Work

There have been significant advancements in frequent pattern mining (FPM) over the past 10 years, as academics have worked to improve algorithmic efficiency, scalability, and practical use in several sectors. Early fundamental work, including the Apriori algorithm, which presented a systematic way to find frequently recurring itemsets through candidate generation, laid the groundwork for the framework. However, Apriori's processing complexity prompted the development of more efficient algorithms, such as Han et al.'s FP-Growth [10][11], which compresses the data using a tree-based structure, eliminating the need for candidate generation and speeding up the mining process [8]. Enhancing these algorithms' performance has been the aim of additional research in order to handle the increasing volumes of data that are common in big data environments.

For instance, the MapReduce framework's distributed computing capabilities is used by Li et al.'s Parallel FP-Growth (PFP) approach to increase scalability and reduce execution times for large datasets [12]. In a similar vein, Borgelt and Kruse's BigFIM approach combines the benefits of Apriori and Eclat with parallelism to improve performance in huge data settings [26]. In addition, frequent pattern mining and high-utility itemset mining (as exemplified by Liu et al.'s HUIMiner) are combined to enhance the applicability of FPM and provide more informative data for decision-making processes by considering the utility or significance of items rather than just their frequency. Developments like the SPMF open-source data mining library by Fournier-Viger et al. have opened up access to sophisticated FPM techniques for a larger audience, facilitating more research and application creation [26].

Moreover, Chen et al.'s improvement of e-commerce recommendation systems is an illustration of the diverse uses of FPM, as is the integration of FPM with machine learning and deep

learning. According to Zhao et al.'s research, adaptive techniques that use reinforcement learning handle the dynamic nature of various data contexts, such as financial markets [17]. All of these advancements show how FPM has progressed from simple approaches to intricate, scalable, and application-specific solutions, highlighting its critical role in finding patterns in intricate, large datasets.

Study	**Data Source**	**Methodology**	**Key Findings**
Yan et al. 2017	Market analysis datasets	Summarizing Frequent Patterns	Clusters similar itemsets for a representative summary, enhancing interpretability and reducing redundancy.
Zaki and Gouda 2016	Various transactional datasets	Eclat with Diffsets	Improves speed and memory efficiency by storing itemset differences, tackling computational bottlenecks.
Aggarwal et al. 2015	Sensor networks, bioinformatics	FPM with Uncertain Data	Handles uncertainty with probability distributions, enabling frequent pattern discovery in noisy data.
Leung et al. 2017	Online retail, social media	Fast and Scalable Top-K FPM	Focuses on speed and scalability for real-time data analysis, suitable for quick pattern detection.
Chen et al. 2020	E-commerce datasets	FPM integrated with Deep Learning	Enhances recommendation systems by improving prediction accuracy of user behavior.
Borgelt and Kruse 2014	Big data environments	BigFIM (Combination of	Balances speed and memory efficiency,

		Apriori and Eclat)	effective for large-scale datasets.
Wang et al. 2020	Big data platforms (e.g., Spark)	Efficient Algorithms in Spark	Utilizes Spark's distributed computing for optimized performance and execution time in big data mining.
Zhao et al. 2018	Real-time financial markets	Adaptive FPM with Reinforcement Learning	Adjusts strategies dynamically based on real-time feedback, improving performance in evolving datasets.
Fournier-Viger et al. 2017	Multiple domains	SPMF (Open-Source Data Mining Library)	Provides a comprehensive toolkit, facilitating frequent pattern mining research and development.

Table 2.1: Related Study in Frequent Pattern Mining

2.3 Overview of Association Rule Mining

Association rule mining is a vital component of frequent pattern mining, which searches large datasets for meaningful relationships between items. Finding patterns and associations that might provide meaningful information is a common use of this technique in a number of domains, including market basket research, internet usage mining, and bioinformatics. In data mining, an association is a link, connection, or union between two or more objects or components. Association rule mining aims to identify relationships between things that often occur together in datasets. The link between the presence of one item and the existence of another is demonstrated by these laws [7]. For example, at a grocery shop, the sales of milk and bread may be tightly correlated, indicating that these products should be placed adjacent.

The common way to express association rules is as X→Y, where X is the itemset on the left side of the rule called antecedent and Y is the itemset on the right side of the rule called consequent [8]. This expression indicates that Y is likely to occur as well if X occurs.

Examples of such rules include:

- Milk → Bread
- Bread → Butter
- Cheese → Bread

In these instances, a client purchasing milk is probably going to purchase bread as well, or a customer purchasing bread may purchase butter as well. Retailers may increase sales by optimizing product placement and marketing techniques with the use of these relationships.

2.3.1 Key Concepts in Association Rule Mining

- **Itemsets and Support:**
  Itemset: A combination of one or more objects.
  Support: The percentage or frequency of transactions in the dataset that have a certain itemset in them. It shows the frequency with which an itemset occurs in the database.

- **Frequent Itemsets**:- If an itemset's support exceeds a minimum support level that the user has determined, it is considered frequent. The creation of association rules is based on these frequently occurring itemsets.

- **Association Rules**:- An implication statement of the form A→B, where A and B are itemsets, is known as an association rule. This rule suggests that if *A* occurs in a transaction, *B* is likely to occur as well.

- **Confidence**:- The likelihood that itemset *B* appears in transactions that contain itemset *A*. It is defined as the ratio of the support of $A \cup B$ to the support of *A*.

- **Lift**:- By comparing the observed support of $A \cup B$ to the predicted support in the event that A and B were independent, lift quantifies the strength of an association rule. A

positive correlation between A and B is shown by a lift value larger than 1.

2.3.2 Applications of Association Rule Mining

- Market basket analysis: This technique helps businesses optimize their inventory and layout strategies by identifying goods that are commonly purchased together.

- Web use mining: Analyzing user behavior to enhance website design and personalize user interfaces.

- Bioinformatics: Finding correlations between genetic markers and illnesses to support medical diagnosis and study.

2.3.3 Benefits and Challenges of Association Rule Mining

Benefits:

- Uncovers hidden patterns and relationships in data.
- Helps in making data-driven decisions in various fields.
- Can handle large datasets efficiently with advanced algorithms.

Challenges:

- Carefully selecting the support and confidence requirements is necessary to avoid creating an excessive or insufficient number of rules.
- May produce a large number of duplicate or useless rules that need to be filtered and post-processed.
- Scalability and efficiency issues may arise when working with high-dimensional or extraordinarily large datasets.

In summary, association rule mining is an effective technique for frequent pattern mining that provides vital details about the relationships between the objects in a dataset [22]. Through the use of algorithms such as Apriori, FP-Growth, and Eclat, people may effectively identify significant trends that facilitate decision-making in many sectors.

2.4 Frequent Pattern Mining Algorithms

2.4.1 Apriori Algorithm

Apriori is the name of the fundamental algorithm used to extract and mine frequent patterns. In 1994, R Agarwal and R Srikant presented it. For it to work, a horizontally laid up database is required. A create and test methodology based on Boolean association rules is utilised. BFS (breadth first search) is used in it. Apriori discovers several k itemsets, from which it constructs a bigger itemset of k+1 items. Before presenting the Apriori for each item, this method first looks through the database to find all frequently recurring things based on support value [1][2][3]. An item's frequency is later calculated by the count of the times it appears in all transactions. Every infrequent item is neglected. It unites two initial phase sets with (n-1) similar elements in the nth pass [14]. The result of the first pass, which begins with a single item, is the candidate set Cn . The second step of the method measures the frequency with which each candidate set occurs, and then it prunes any itemset that is used seldom. The algorithm ends when there are no further extensions[17].

Apriori employs a two step process:

- **Join Step:** Lk−1 is joined with itself to generate Ck.
- **Step of Pruning:** A non-frequent (k−1)-itemset cannot be a subset of a frequently occurring k-itemset.

Apriori Frequent Itemset Algorithm

INPUT: An item basket file D with a support threshold σ

OUTPUT: An itemset list F(D, σ)

METHOD:

1: C1 ← {{i} | i ∈ J}

2: k ← 1

3: while Ck ≠ {} do

```
4:    # Compute the supports of all candidate itemsets
5:    for all transactions {tid, I} ∈ D do
6:        for all candidate itemsets X ∈ Ck do
7:            if X ⊆ I then
8:                X.support++
9:   # Extract all frequent itemsets
10:   Fk ← {X | X.support ≥ σ}
11:   # Generate new candidate itemsets
12:   Ck+1 ← {}
13:   for all X, Y ∈ Fk do
14:       if X[1:(k-1)] = Y[1:(k-1)] and X[k] < Y[k] then
15:           I ← X ∪ {Y[k]}
16:           if ∀J ⊂ I, |J| = k: J ∈ Fk then
17:               Ck+1 ← Ck+1 ∪ {I}
18:   k++
19: return ∪k Fk
```

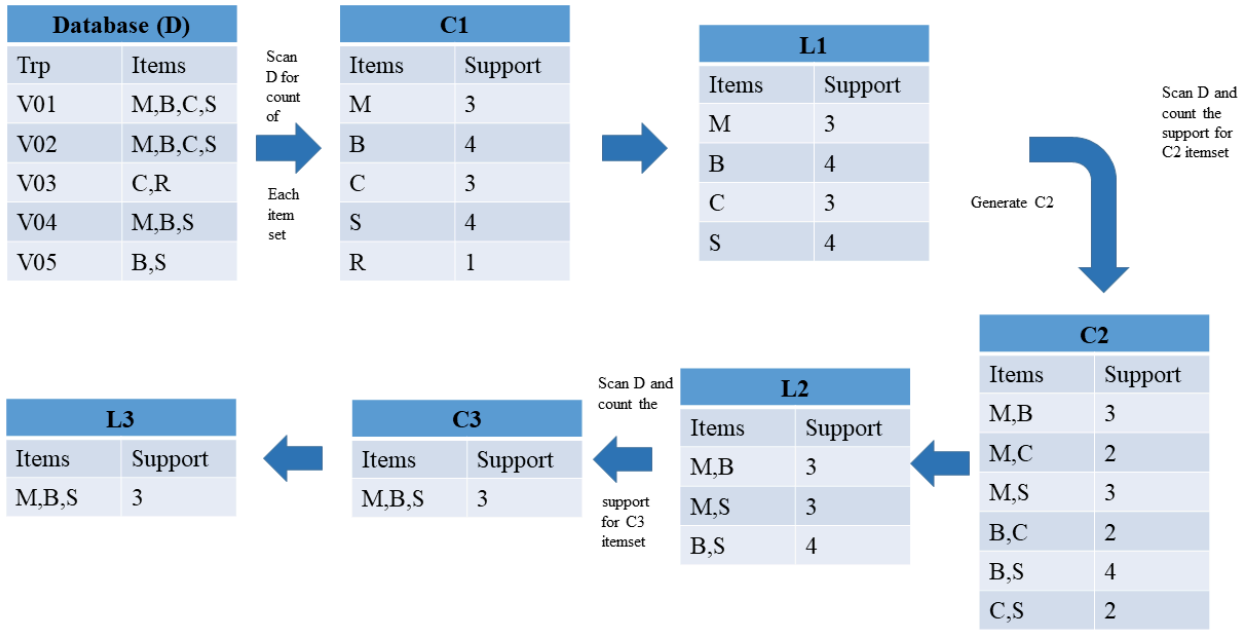


Fig. 2.1. Example of Apriori Algorithm.

As an instance, let's look at the approach shown in Fig. 3 for creating candidate itemsets (C) and frequent itemsets (L) with a minimum support of 60% (or three transactions). The method first looks for item support in the database and removes things that don't fulfill the minimal support needed to get L1. It then uses self-join operations to create C2 from L1, then it repeats the filtering and generation step to create C3 from L2, and finally it extracts L3 from C3. Items that occur often are represented by the frequent itemsets that arise, such as M, B, and S.

2.4.2 Frequent Pattern - Growth Algorithm

Introduced in 2000 by Han, Pei, and Yin, the FP-Growth algorithm revolutionized frequent pattern mining by eliminating the need for candidate generation entirely [4]. This approach enables the mining of frequent itemsets without the necessity of generating candidate itemsets. An approach called Frequent Pattern Growth FP-Growth extract frequent itemset from given data without utilizing a costly candidate itemset generation step. In order to convert frequent items into a Frequent pattern tree divide-and-conquer strategy is used. The Frequent Pattern -Tree is then separated into a group of Conditional FP-Trees in order to mine each item independently [4][10]. By continuously scanning through subdivided Conditional FP-Trees, this algorithm finds large, repeated patterns[11]. Figure 4 displays the Conditional FP-Tree connected to nodeI3, and

Table 2 provides information on each Conditional FPTree shown in Figure 3. A "sub-database" called the Conditional Pattern Base contains all prefix paths in the Frequent Pattern Tree that also co-exist with every frequent length 1 item[25]. The Conditional FP Tree is built using it, it also generates all the repeated patterns associated with each frequent length 1 items. FP- Growth is a two-step approach.

FP-Growth Algorithm

INPUT: A file D with item baskets, an item prefix I such that $I \subseteq J$, and a support threshold σ

OUTPUT: An itemsets list F[I](D, σ) for the given prefix

METHOD:

1: $F[I] \leftarrow \{\}$

2: for all $i \in J$ occurring in D do

3: $F[I] \leftarrow F[I] \cup \{I \cup \{i\}\}$

4: Di ← {} # Create Di

5: $H \leftarrow \{\}$

6: for all $j \in J$ occurring in D such that $j > i$ do

7: if support($I \cup \{i, j\}$) $\geq \sigma$ then

8: $H \leftarrow H \cup \{j\}$

9: for all (tid, X) $\in$ D with $I \in X$ do

10: $Di \leftarrow Di \cup \{(tid, X \cap H)\}$

11: Compute $F[I \cup \{i\}](Di, \sigma)$

12: $F[I] \leftarrow F[I] \cup F[I \cup \{i\}]$

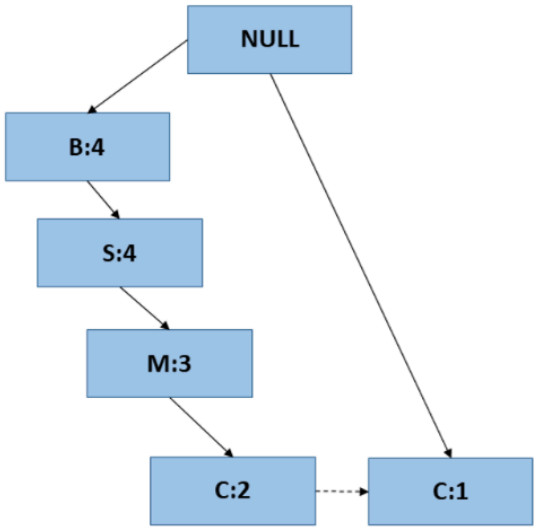
13: return F[I]

Database (D)	
Trp	Items
V01	M,B,C,S
V02	M,B,C,S
V03	C,R
V04	M,B,S
V05	B,S

Ordered Items	
Trp	Items
V01	B,S,M,C
V02	B,S,M,C
V03	C
V04	B,S,M
V05	B,S

Frequent Items	
Items	Support
B	M,B,C,S
S	M,B,C,S
M	C,R
C	M,B,S

(a) During first scan of Database (D)



(b) Frequent Pattern Tree after second scan

Items	Conditional pattern base	Conditional FP-tree	Frequent Pattern Generated
C	(B,S,M:2)	-	-
M	(B,S:3)	(B,S:3)	(B,S,M:3), (B,M:3),(S,M:3)
S	(B:4)	(B:4)	(B,S:4)

(c) Frequent Itemset Generation

Fig. 2.2. Example of FP-Growth Algorithm.

For instance, consider the process of generating frequent itemsets and constructing an FP-tree with A minimum support threshold of 60%, which corresponds to at least 3 transactions, as depicted in Fig-4. In (a), during the initial scan of dataset D, the first frequent items are identified, followed by arranging the items in the original database to obtain ordered itemsets in decreasing order of support. Subsequently, in (b), to create the FP-Tree, the database is examined once again. Finally, in (c), Frequent patterns are formed utilizing both the conditional pattern

base and FP-tree. Only frequent patterns with three or more transactions meeting the minimum support count of 60% are considered, such as (B, S, M: 3), (B, M: 3), (S, M: 3), and (B, S: 4).

2.4.3 Eclat Algorithm

The ECLAT (Equivalence Class Clustering and bottom-up Lattice Traversal) algorithm, introduced by Zaki in 2000, is a prominent method in pattern mining, specifically for frequent itemset mining. Unlike the breadth-first search strategy used by the well-known Apriori algorithm, ECLAT employs a depth-first search approach, making it particularly efficient for dense datasets [6]. ECLAT begins by generating transaction ID (TID) lists for all single items, associating each item with a list of transactions where it appears. The algorithm then recursively processes these itemsets, intersecting TID lists of subsets to form new itemsets, and determines their support by the length of these TID lists [9]. Itemsets are considered frequent if they reach or surpass the minimal support criterion., while those that do not are pruned to reduce the search space. This method leverages the TID list representation to perform swift intersection operations, making it memory-efficient and fast, especially in contexts with dense data [10]. Despite its efficiency, ECLAT's depth-first search can be less effective for sparse datasets and may result in a more complex implementation.

ECLAT Algorithm

INPUT: A database Q with item baskets, an item prefix I such that $I \subseteq J$, and a support threshold T

OUTPUT: A list of itemsets for the given prefix F[I](Q, T)

METHOD:

1: $F[I] \leftarrow \{\}$

2: for all $i \in J$ occurring in Q do

3: $F[I] \leftarrow F[I] \cup \{I \cup \{i\}\}$

4: $Qi \leftarrow \{\}$, $H \leftarrow \{\}$ # Create Qi

5: for all $j \in J$ occurring in Q such that $j > i$ do

6: if support(I ∪ {i, j}) ≥ T then

7: H ← H ∪ {j}

8: for all (tid, A) ∈ Q with I ⊆ A do

9: Qi ← Qi ∪ {(tid, A ∩ H)}

10: Compute F[I ∪ {i}](Qi, T)

11: F[I] ← F[I] ∪ F[I ∪ {i}]

12: return F[I]

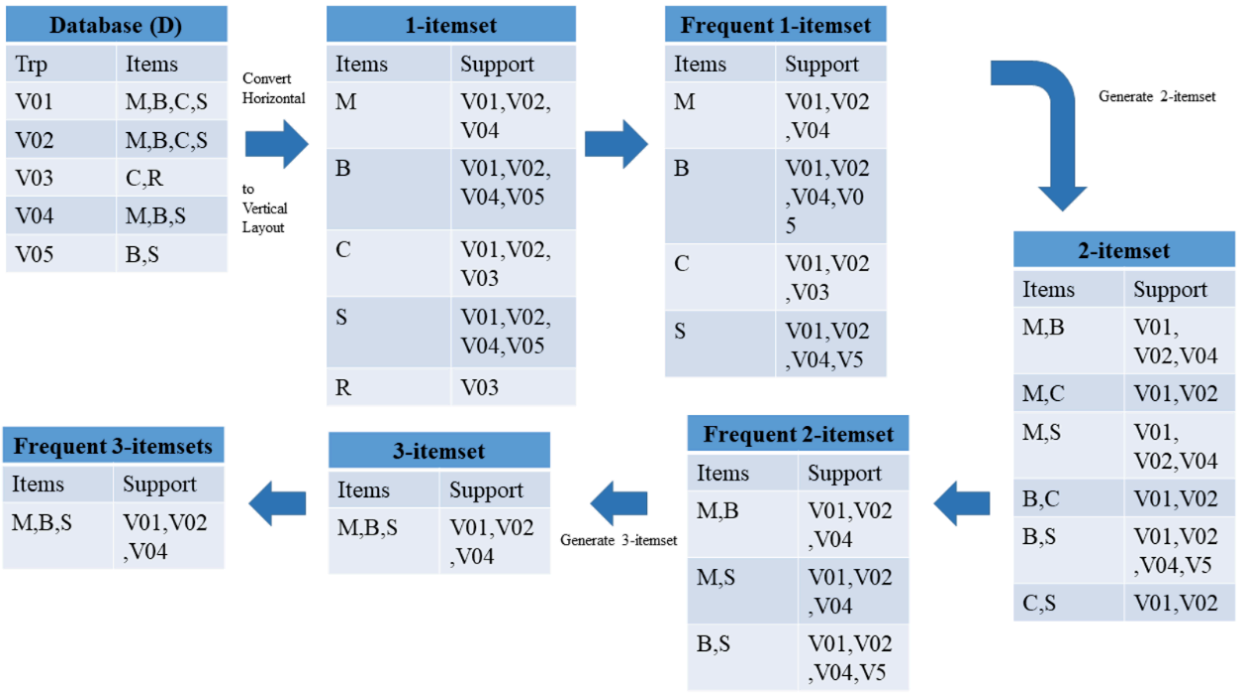


Fig. 2.3. Example of ECLAT Algorithm.

As an illustration Take a look at Fig. 5, which shows the process of creating itemsets and frequent itemsets with a minimum support count of 60% (or three transactions). To produce frequent 1-itemsets, the dataset D is first scanned to transform its horizontal to vertical arrangement. Items that do not fulfill the minimal support criteria are then removed. An

intersection approach is then used to create 2-itemsets from the frequently occurring 1-itemsets. Items are then removed from the 2-itemsets, and the resultant frequent 2-itemsets are used to create 3-itemsets. Lastly, from the 3-itemsets, frequent 3-itemsets are derived, indicating that items M, B, and S are commonly seen.

CHAPTER 3

DATASETS AND METHODOLOGY

3.1 Datasets Description

3.1.1 Chess Dataset

The chess dataset from UCI repository is a classic benchmark for assessing frequent itemset mining algorithms because of its high dimensionality and hierarchical complexity. It contains 3,196 instances and 37 binary attributes that represent many possible chess endgame scenarios. Each attribute is either set to zero or one, indicating whether a piece or set of circumstances is present on the chessboard. Even though the database is small, every characteristic it has corresponds to an entirely different facet of game state thus making it information rich.

Such empirical studies can be conducted using "A Performance based Empirical Study of the Frequent Itemset Mining Algorithms that was cited in IEEE ICPCSI-2017 where the chess dataset plays an important role. Eclat uses depth first search approach which makes it faster than Apriori that applies breadth first search strategy that leads to much memory use and numerous database scans. On the other hand, FP-Growth achieves fast itemset mining without explicitly generating candidate sets by compressing a tree structure.

This unusual combination of small size mixed with high complexity suggests that this dataset has significant implications about how well algorithms behave in practice and allows for computational experimentation.

3.1.2 Accident Dataset

The data set about accidents was given from the Frequent Itemset Mining Dataset Repository. There are a total of 340,183 records in this dataset each describing a single accident. It is an extensive collection meant to assess frequent traditional itemset mining algorithms. Each record has several binary attributes indicating whether certain conditions or rules regarding the incidents are present or not. Vital features include the place of occurrence of the accident, which helps identify areas prone to accidents; weather at scene, which details if it was clear, rainy or foggy and it is important for analyzing how weather affects accident rates; and condition of road, which

tells if it was dry, wet or icy, thus helping understand how accidents depend on status of roads. Also different vehicle-specific patterns could be studied due to the list of involved vehicle types and severity field being an indicator that ranges from minor to fatal and plays a vital role in showing these variables responsible for severe crashes. In order to find patterns and relationships between these attributes such as common combinations of weather and traffic conditions that commonly result in accidents the dataset is used in frequent itemset mining. Because of this, the information is priceless for analyzing traffic safety, formulating laws, and enhancing traffic and automobile safety protocols.

3.2 Methodology

3.2.1 Problem Statement

One of the primary objectives in data mining is frequent pattern mining, which concentrates on extracting recurrent itemsets from transactional datasets. Finding subsets of elements that commonly co-occur throughout transactions is the main goal in order to uncover underlying patterns and correlations within the data. Formally speaking, the task is to identify all itemsets I for which the support (i.e., the percentage of transactions containing I) exceeds a certain threshold, denoted as min_support.

Given a transactional database D comprising N transactions and a minimum support threshold min_support, the problem can be formulated as follows:

Input:

D: Transactional database containing N transactions.

min_support : Minimum support threshold.

Output:

Frequent itemsets: Set of all itemsets I where the support of I exceeds min_support.

Finding these frequent itemsets from the transactional database is the main goal of frequent pattern mining. The itemsets that have been discovered contribute as fundamental components for further studies, such as the creation of association rules, pattern identification, and market

basket analysis. Effectively mining recurrent patterns from large transactional databases requires overcoming several challenges, such as issues with computational complexity, scalability, and algorithmic efficiency. Conventional methods such as FP-growth and Apriori may not scale well when used to datasets with millions of transactions. This requires further research and optimisations.

A systematic approach that incorporates optimisation approaches, algorithm selection, and parameter adjustment is required to overcome these challenges. The trade-offs between processor power, memory use, and pattern quality must be managed by academics and practitioners in order to provide useful mining results within reasonable time constraints.

3.2.2 Proposed Method

Transaction Aggregation

To make frequent pattern mining more efficient and effective, transaction aggregation is applied in this research. With this technique, the individual transactions are joined together based on their commonality like the product category that they belong to, time frame or customer id and then summarized into just a few aggregated records. These aggregated records represent the transactional data more briefly but at the same time maintain the vital information required for pattern mining.

Transaction aggregation has reduced dataset size to within handleable limits by associations of interest in large datasets for frequent pattern mining algorithms. By combining many transactions into one aggregated item, the total dataset size is significantly reduced thus speeding up processing and using less memory. This is particularly important when dealing with large scale transaction datasets where traditional methods of pattern mining would find it hard to cope effectively with such huge quantities of information.

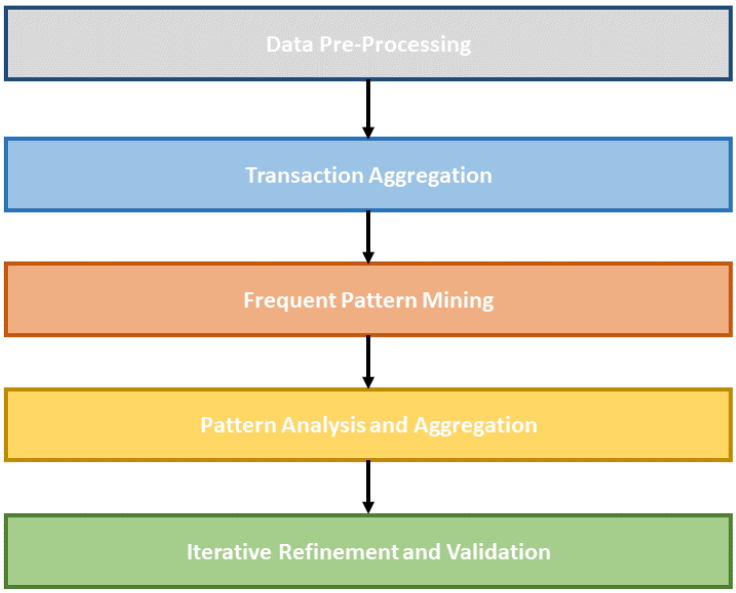


Fig.3.1. Flowchart of Proposed Method.

In fact, the transaction aggregation has indeed many practical applications and its usage will depend on the specific requirements of a given study. For instance, to analyze buying patterns with respect to individual customers transactions can be aggregated by customer ID.? In addition, to reveal seasonality and time trends transactions may be aggregated by time period. Moreover, market basket dynamics and product affinities can be understood if one aggregates transactions by product category. What determines the choice of aggregations criteria are research objectives, domain expertise and dataset properties.

To apply transaction aggregation in this case, we can treat two or more chess records consisting of similar moves and values as one record. For instance, two grocery purchases amounting to the same products can be treated as one integrated transaction. As such, this consolidation method eliminates duplication while preparing the dataset for further analysis thereby making it possible to carry out more successful frequent pattern mining.

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 Algorithms Used and Performance Measure

Three well-known frequent pattern mining algorithms - Apriori, FP-Growth, and ECLAT were used in our study on two different datasets Accidents, and Chess. Our main goal was to assess and contrast these algorithm's performances using two crucial parameters: memory use and execution time. Megabytes are used to measure memory, whereas milliseconds are used to measure time. The evaluation of the effect of this preprocessing step on time and memory use is done by comparing the results before and after transaction aggregation. It was anticipated that Apriori, which is renowned for its simple candidate generation method, would demonstrate increased temporal complexity, particularly when dealing with bigger datasets. FP-Growth sought to show more effective memory utilization and quicker processing times by doing away with the requirement for candidate creation by using a tree-based structure. ECLAT was evaluated for its ability to balance memory usage and time efficiency by utilizing intersection procedures and vertical data format. We attempted to explain each method's advantages and disadvantages while working with datasets that differed in quality, therefore offering a better understanding of how well each algorithm would work in various data mining situations.

4.2 Result Evaluation

During the experiment, various threshold values for minimum support (min_sup) were considered between the ranges 0.6 to 0.9 (inclusive) on all the datasets before and after data preprocessing. Especially when dealing with all the data sets , transaction aggregation is essential for streamlining the data preparation stage. To cut down on repetition and improve algorithm performance, related transactions are grouped together in transaction aggregation.

- The time and memory utilization of the algorithm running on the chess dataset are displayed in the following visual. Using 3196 transactions in the dataset, the algorithm found 254944 common itemsets in total. According to the analysis, the Eclat method used more memory resources to mine frequently occurring itemsets, but the Apriori technique took longer to finish its execution before preprocessing.

  In comparison to that the depicted chart reflects the outcomes of the algorithm applied to the preprocessed chess dataset, where transaction aggregation was performed. After preprocessing, the total number of transactions reduced to 3056, while the total number of frequent itemsets remained the same at 254944. It's observed that the Apriori algorithm still took more time to execute compared to other algorithms, while Eclat continued to utilize more memory resources for mining frequent itemsets.

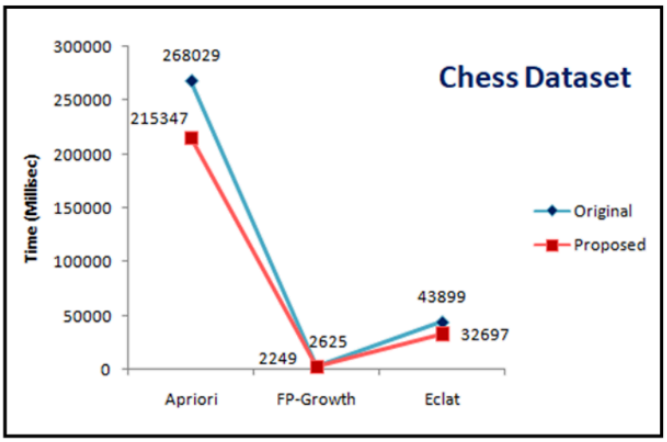
Table 4.1: Result of chess dataset on various support counts before transaction aggregation.

Support	**Time (millisec)**		
	Apriori	**FP-Growth**	**Eclat**
0.9	4000	520	3572
0.85	13050	890	8956
0.8	20600	1174	10168
0.75	28371	1229	13258
0.7	50268	1383	29700
0.65	139170	1820	36640
0.6	268029	2625	43899

Table 4.2: Result of chess dataset on various support counts after transaction aggregation.

Support	Time (millisec)		
	Apriori	**FP-Growth**	**Eclat**
0.9	3648	500	3260
0.85	11307	746	7488
0.8	18816	1007	9375
0.75	24690	1183	11328
0.7	48379	1306	26527
0.65	127816	1618	30065
0.6	215347	2249	32697

The table shows the three common pattern mining algorithms Aprori, FP-Growth, and ECLAT execution times (in milliseconds) over various support thresholds. All algorithms have longer execution times when the support threshold drops, which means that more itemsets are seen as frequent. This is because there is a greater search space. Apriori's extensive candidate generation procedure causes it to handle huge datasets inefficiently, as seen by its greatest execution times. Because FP-Growth uses an effective tree-based strategy instead of directly creating candidates, it regularly performs better than Apriori and shows quicker execution times. ECLAT uses vertical data formats to balance memory utilization and execution time, resulting in intermediate performance that is typically quicker than Apriori but slower than FP-Growth.



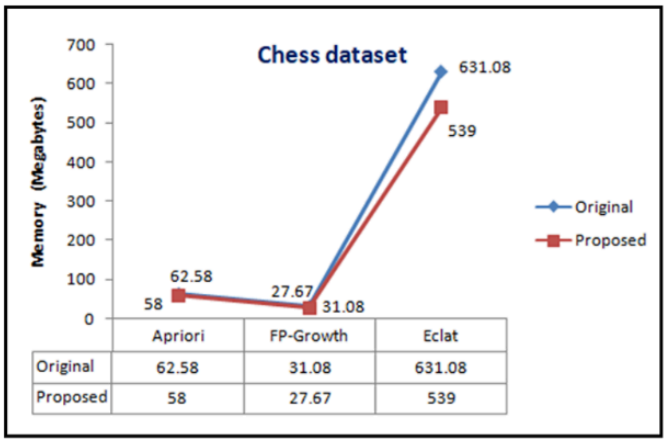


Fig.4.1. a) Execution time of Original vs Proposed model for support 60%.

b) Memory Usage of Original vs Proposed model for support 60%.

The chart above shows the runtime and memory consumption of the original and suggested models for the Apriori, FP-Growth, and ECLAT algorithms at a 60% support level. All three methods demonstrate a noticeable reduction in runtime and memory use after the application of transaction aggregation in the suggested paradigm. This comparison highlights the effectiveness of transaction aggregation in optimizing frequent pattern mining algorithms.

- The results of running all three algorithms on the Accident dataset are shown in the chart below. There are 7750 common itemsets out of the 89420 transactions in the dataset. Interestingly, when mining frequently occurring itemsets, the Eclat method used more memory while the Apriori technique used more time before data preprocessing.

  Comparative chart shows the outcomes of applying the algorithms on the Accident dataset after transaction aggregation was used for data preprocessing. After processing 89420 transactions, 7750 frequent itemsets were found in this improved dataset. Remarkably the Apriori method continued to show higher time consumption for the mining of frequent itemsets, while the Eclat approach continued to show higher memory usage. This was even when the transactions were aggregated.

Table 4.3: Result of Accident dataset on various support count before transaction aggregation

Support	Time (millisec)		
	Apriori	FP-Growth	Eclat
0.9	5620	675	3990
0.85	9748	973	8832
0.8	13429	1237	11927
0.75	25120	1458	26402
0.7	36437	2047	33195
0.65	47018	2310	48361
0.6	93891	3250	52843

Table 4.4: Result of Accident dataset on various support counts after transaction aggregation.

Support	Time (millisec)		
	Apriori	FP-Growth	Eclat
0.9	5400	450	3600
0.85	9074	700	7967
0.8	13376	967	9438
0.75	23962	1158	14822
0.7	36437	1346	19894
0.65	47019	1973	24128
0.6	93891	2849	39635

The table displays the milliseconds (ms) of execution time for three popular pattern mining algorithms: Apriori, FP-Growth, and ECLAT, for different support levels after transaction aggregation. We successfully decreased the dataset size by combining similar rows, which

enhanced the overall algorithmic efficiency. Because of its candidate generating mechanism, Apriori demonstrated the longest execution durations even after this optimization. Thanks to its tree structure, FP-Growth was able to continuously record the lowest execution times, indicating that it was efficient at processing the combined transactions. ECLAT used its vertical data format to achieve an intermediate performance, slower than FP-Growth but quicker than Apriori. These findings emphasize the role that transaction aggregation plays in improving the effectiveness of frequent pattern mining, emphasizing the improved performance of FP-Growth in particular.

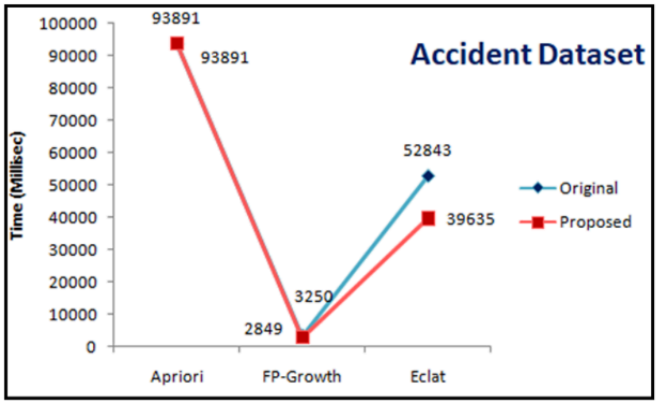
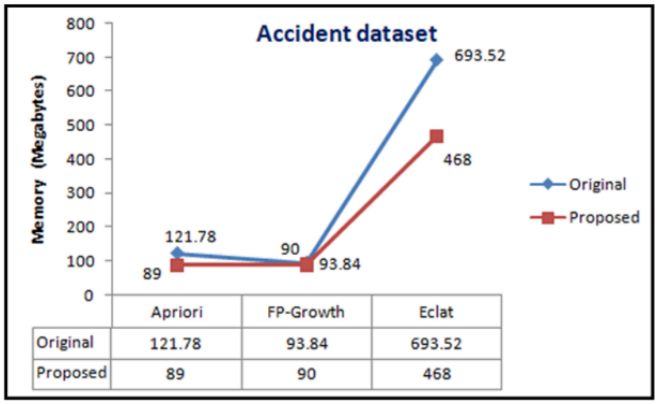


Fig.4.2. a) Execution time of Original vs Proposed model for support 60%.



b) Memory Usage of Original vs Proposed model for support 60%.

The chart above shows the runtime and memory consumption of the original and suggested models for the Apriori, FP-Growth, and ECLAT algorithms at a 60% support level. All three algorithms demonstrate a noticeable reduction in runtime and memory use after the application of transaction aggregation in the suggested paradigm. This comparison highlights the effectiveness of transaction aggregation in optimizing frequent pattern mining algorithms.

CHAPTER 5

CONCLUSION

A thorough examination of all datasets, both pre and post data preprocessing, especially via transaction aggregation, yields important information on how well frequent itemset mining methods work. Between all datasets, the Apriori method always took longer time even after aggregating the transaction, whereas the Eclat algorithm always used more memory. This trend highlights the computational trade-offs that come with comparing Eclat's emphasis on memory optimisation versus Apriori's prioritization on time efficiency. Nevertheless, the implementation of transaction aggregation resulted in a visible enhancement of algorithmic performance, as demonstrated by decreased time and memory use.

The post-preprocessing results show that transaction aggregation successfully simplified datasets enabling better mining of frequent itemset. This indicates that transaction aggregation is an important preprocessing methodology for improving the efficiency of frequent itemset mining methods hence allowing quicker study of large databases with lesser use of memory. On a general note, such findings emphasize on the importance of data preprocessing techniques in the optimization of frequent itemset mining algorithms for wider use applications.

REFERENCES

[1] R. Agrawal, T. Imielinski & A. Swami Mining. 1993. “Association Rules between Sets of Items in Large Databases “,Proceedings of the 1993 ACM SIGMOD Conference, pp.1-10

[2] Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216.

[3] R. Agrawal, R. Srikant et al., “Fast algorithms for mining association rules,” in Proceedings of the 20th international conference of very large data bases, VLDB, vol. 1215, 1994, pp. 487–499.

[4] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in ACM SIGMOD Record, vol. 29, no. 2. ACM, 2000, pp. 1–12.

[5] Sagar Bhise1,Prof. Sweta Kale,”Efficient Algorithms to find Frequent Itemset Using Data Mining”, © 2017, IRJET | Impact Factor value: 5.181 | ISO 9001:2008 Certified Journal, -ISSN: 2395-0072.

[6] M. J. Zaki, "Scalable Algorithms for Association Mining," in Proceedings of the 16th International Conference on Data Engineering (ICDE), 2000, pp. 589-590.

[7] Anil Vasoya a, Dr. Nitin Koli, “Mining of association rules on large database using distributed and parallel computing”, 1877- 0509 © 2016 blished by Elsevier B.V Procedia Computer Science 79 (2016) 221 – 230.

[8] Meera Narvekar, Shafaque Fatma Syed, “An Optimized Algorithm for Association Rule Mining Using FP Tree”, Procedia Computer Science 45 (2015) 101 – 110, 1877-0509 © 2015 The Authors. Published by Elsevier B.V.

[9] Xiaofeng Zheng , Shu Wang, “Study on the Method of Road Transport Management Information Data Mining Based on Pruning Eclat Algorithm and MapReduce”, Procedia -

Social and Behavioral Sciences 138 (2014) 757 – 766, 1877-0428 © 2014 Published by Elsevier Ltd.

[10] JeffHeaton,"Comparing Dataset Characteristics that Favor the Apriori, Eclat or FP-Growth Frequent Itemset Mining Algorithms", 978-1-5090- 2246-5/16 © 2016

[11] Luca Cagliero; Paolo Garza, "Infrequent Weighted Itemset Mining Using Frequent Pattern Growth", IEEE Transactions on Knowledge and Data Engineering (Volume: 26, Issue: 4, April 2014) Pages: 903 - 915, DOI: 10.1109/TKDE.2013.69, Print ISSN: 1041-4347.

[12] M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li et al., "New algorithms for fast discovery of association rules," in KDD, vol. 97, 1997, pp. 283– 286.

[13] Sagar Bhise1,Prof. Sweta Kale,"Efficient Algorithms to find Frequent Itemset Using Data Mining", © 2017, IRJET | Impact Factor value: 5.181, ISO 9001:2008 Certified Journal, -ISSN: 2395-0072.

[14] K. Geetha, Sk. Mohiddin, "An Efficient Data Mining Technique for Generating Frequent Item sets", International Journal of Advanced Research in Computer Science and Software Engineering 3(4), April - 2013, pp.571-575.

[15] Tuong Le, Bay Vo, ―An N-list-based algorithm for mining frequent closed patterns‖, Expert Systems with Applications, Volume 42, Issue 19, Pages 6648-6657, 1 November 2015.

[16] Zhi-Hong Deng, Sheng-Long Lv, ―Fast mining frequent itemsets using Nodesets‖, Expert Systems with Applications, Volume 41, Issue 10, Pages 4505-4512, August 2014.

[17] Xiang Cheng, Sen Su, Shengzhi Xu, Zhengyi Li, "DP-Apriori: A differentially private frequent itemset mining algorithm based on transaction splitting", Computers & Security, Volume 50, Pages 74-90, May 2015.

[18] J. Han, and M. Kamber, "Data Mining concepts and techniques", Elsevier Inc., Second Edition, San Francisco, 2006..

[19] L. Shi, J. N. Bai, and Y. l. Zhao, “Mining association rules based on apriori algorithm and application,” In Proc. of IFCSTA, vol. 3, pp. 141– 145, Dec 2009.

[20] M. Shaheen, M. Shahbaz, and A. Guergachi, “Context based positive and negative spatio-temporal association rule mining,” Knowl.-Based Syst., vol. 37, pp. 261–273, 2013.

[21] D. H. Setiabudi,G. S. Budhi and W. J. Purnama, “Data mining market basket analysis' using hybrid-dimension association rules, case study in Minimarket X”, in Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on,2011,Page(s):196-199.

[22] S. Tao and P. Gupta, “Implementing improved algorithm over apriori data mining association rule algorithm,” IJCST, vol. 3, pp. 489 – 493, Jan-Mar 2012.

[23] Anindita Borah and Bhabesh Nath, “Comparative evaluation of pattern mining techniques: an empirical study”, Complex & Intelligent Systems, volume 7, Issue 2, 589-619 (2021).

[24] M. Hahsler, B. Gruen, and K. Hornik, “arules – A computational environment for mining association rules and frequent item sets,” Journal of Statistical Software, vol. 14, no. 15, pp. 1–25, October 2005. [Online]. Available: http://www.jstatsoft.org/v14/i15/

[25] Syed Khairuzzaman Tanbeer, Chowdhary Farhan Ahmed, Byeong-Soo Jeong and Y.W.Lee, CP-Tree: A Tree Structure for Single-Pass Frequent Pattern Mining ,In proceedings of 12th Pacific Asia Conference,2008.

[26] C. Borgelt. An Implementation of the FP- growth Algorithm. Proc. Workshop Open Software for Data Mining (OSDM’05 at KDD’05, Chicago,IL),1– 5.ACMPress, New York, NY, USA 2005.

ANNEXURE-IV

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Unveiling Hidden Patterns: Exploring the potential of frequent mining algorithm Total Pages 30 Name of the Scholar Sonali Rani Supervisor (s)

(1) Dr. Ritu Agarwal

(2)_____

(3)_____

Department INFORMATION TECHNOLOGY, DELHI TECHNOLOGICAL UNIVERSITY

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used:Turnitin Similarity Index:14% , Total Word Count:6330

Date:30/05/2024

Candidate's Signature **Signature of Supervisor(s)**

PAPER NAME
SonaliThesis.pdf

AUTHOR
Sonali Rani

WORD COUNT
6330 Words

CHARACTER COUNT
35142 Characters

PAGE COUNT
30 Pages

FILE SIZE
987.7KB

SUBMISSION DATE
May 30, 2024 10:01 AM GMT+5:30

REPORT DATE
May 30, 2024 10:02 AM GMT+5:30

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- 8% Publications database
- Crossref database
- Crossref Posted Content database
- 11% Submitted Works database

Excluded from Similarity Report

- Bibliographic material