Sana'a University Journal of Applied Sciences and Technology https://journals.su.edu.ye/index.php/jast
Vol. 1 | No. 4 | Page 358 – 367 | 2023 |
ISSN: 2958-9568
Doi: 10.59628/jast.v1i4.663

Analysis Of Mining Algorithms for Patterns of Frequent Item sets and Hidden

Abdulkader Mohammed Abdulla Al-Badani[1,*], Abdualmajed A. G. Al-Khulaidi[1]
[1] Department of Computer Science, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen.
***Corresponding authors Emails:** abdulkaderbadani@su.edu.ye and alkhulaidi@su.edu.ye

ARTICLE INFO
Article history:
Received: Oct 7, 2023
Accepted: Nov 9, 2023
Published: Nov, 2023

KEYWORDS
1. Data mining
2. frequent patterns
3. Big data
4. FP-growth
5. KDD

ABSTRACT
Frequent set-item mining is a widely used technique in the field of data mining. The algorithm utilized in Association Rules constitutes the primary source of contention; this approach is memory-intensive and time-intensive. The extraction process for concealed patterns of frequent item sets, on the other hand, becomes more time-consuming as the volume of data grows. Therefore, the necessary algorithm for mining the patterns of frequently occurring concealed item sets must be memory-efficient and fast-running. By analyzing various algorithms for locating frequent item sets, this paper aims to contribute to the development of a more efficient algorithm in this domain. The paper will evaluate the efficiency of different algorithms by considering their memory usage and runtime. Additionally, it will explore potential improvements to existing algorithms to enhance their performance in mining frequently occurring concealed item sets.

CONTENTS

1. Introduction:

Data mining pertains to the algorithmic exploration of vast quantities of data in an attempt to uncover concealed information [1]. The current maximum size of big data in a single data storage unit is several terabytes; this represents a rate of growth that exceeds the processing capacity of current software systems capable of capturing, storing, managing, processing, and analyzing enormous data chunks promptly [2]. Data mining has garnered significant interest within the information industry in recent times. The primary rationale is the substantial volume of data that must be promptly converted into actionable insights and knowledge. This need for quick conversion is driven by the fact that data mining allows businesses to uncover patterns, correlations, and trends that can be used to make informed decisions and gain a competitive edge. Additionally, data mining techniques enable organizations to identify anomalies or outliers in the data, which can be crucial for detecting fraud or identifying potential risks.

Critical decisions are generated by employing data mining techniques, including pattern mining and rule mining [3], to extract intriguing patterns from databases. In transactional databases, two significant procedures known as frequent item-set mining (FIM) [6] and association rule mining (ARM) [4], [5] are utilized to identify intriguing relationships between items. The purpose of

ARM is to extract inviolable principles from a database through the implementation of interestingness measures. Items are frequently summoned in FIM when the anticipated support exceeds the minimum support (minsupp) value specified by the user. To ascertain the relationship between objects, ARM examines binary transaction information for the most robust principles.

An essential area of study [7] is the implementation of association rule technology to design an intrusion detection system that is both efficient and capable of distinguishing between known and unknown attack patterns. Association rule mining (ARM) is a powerful technique used to discover interesting relationships and patterns in large datasets. By analyzing binary transaction information, ARM can identify the most significant principles that govern the occurrence of items. One important application of ARM is in the field of intrusion detection systems, where it can be used to detect and classify both known and unknown attack patterns efficiently. This implementation of association rule technology plays a crucial role in designing effective intrusion detection systems for ensuring network security.

One of the primary methodologies employed in data mining is association rule mining [8], [9], and [10]. It is utilized to identify recurring patterns and associations in transactional databases and other repositories. Its primary applications include bioinformatics, retail, agriculture, innovation, and marketing. The primary objective of association discovery is to identify noteworthy relationships between collections of items from a given database. There are numerous algorithms proposed for locating frequent patterns [11]. Apriori algorithms are one, while the FP growth approach is the other. The apriori method generates patterns of length p+1 by utilizing the prevalent patterns of length p. To generate lengthy and frequent item sets, a significant number of database searches are necessary. Numerous investigations [12], [13], [14], [15], [16], [15], [16], and [17] have implemented the apriori approach. The FPtree is utilized to store the database in the FP Growth method to identify frequent patterns. This approach involves scanning the database only twice [18], [19]. It is thus more rapid than Apriori. The FP-growth approach has numerous alternatives and extensions [20], [28], [29], [30], and [31]. Some of the alternatives and extensions of the FP-growth approach include Eclat [20], FPGrowthPlus [21], FPMax [22], and FPGrowthHUI [23]. These variations aim to improve the efficiency and effectiveness of frequent pattern mining in different scenarios. Additionally, some studies have proposed hybrid approaches that combine the strengths of both Apriori and FP-growth methods [24], [25], [26], and [27].

The objective of this article is to aid in the advancement of a more effective algorithm within this field. The paper will assess the performance of various algorithms in terms of memory consumption and execution time. Additionally, the article will analyze the impact of different data structures on algorithm efficiency. It aims to provide insights into optimizing algorithms for better resource utilization and faster execution in practical applications.

The structure of the remainder of the document is as follows: There is an analysis of existing algorithms in Section 2. The conclusion is in Section 3.

1.1 Data Mining Techniques

Data mining is a component of the Knowledge Discovery in Databases (KDD) process, which also includes the discovery of an algorithm that classifies data patterns based on their applicability to data analysis [32]. Data mining involves the extraction of valuable insights and patterns from large datasets, enabling organizations to make informed decisions and predictions. It plays a crucial role in various fields, such as business, healthcare, and finance, by uncovering hidden relationships and trends that may not be apparent at first glance.

Knowledge or information required for business decision-making is extremely limited, despite the exponential growth of data storage. Data mining is sometimes referred to as "knowledge." This is because data mining involves not only extracting information from large datasets but also transforming it into actionable knowledge that can drive strategic decision-making. By leveraging advanced algorithms and statistical techniques, data mining enables organizations to go beyond surface-level observations and gain deeper insights into their operations, customer behavior, and market trends. Ultimately, the goal of data mining is to empower businesses with the intelligence they need to stay competitive and thrive in today's data-driven world.

The extracted knowledge enables the prediction of present and future behavior. This enables business proprietors to formulate informed and constructive decisions. Data mining is utilized in numerous sectors, including education and aerospace.

Through the application of pattern recognition, statistical, and mathematical techniques, knowledge is extracted from historical data; this knowledge takes the form of facts, trends, associations, patterns, anomalies, and exceptions. There are certain domains in which data mining will be implemented.

These domains include marketing, finance, healthcare, and customer relationship management. By analyzing large datasets, businesses can identify market trends, predict consumer behavior, optimize financial strategies, improve patient outcomes, and enhance customer satisfaction. Data mining plays a crucial role in helping businesses stay competitive and make data-driven decisions in today's fast-paced and data-driven world.

1.2 Knowledge Discovery Process From Data

In essence, the KDD process consists of data mining techniques used to identify patterns in data. The objective of each method is distinct, which completely determines the outcome of the KDD process. These techniques can include classification, clustering, association rule mining, and anomaly detection, among others. By applying these methods, the KDD process aims to extract valuable knowledge and insights from large datasets for various purposes, such as decision-making, prediction, and optimization.

Data Pre-processing:

For the mining procedure, real-time data is generated via data preprocessing. This data preprocessing involves cleaning and transforming raw data into a format suitable for analysis. The real-time data generated is then used for various mining techniques, such as pattern recognition and predictive modeling.

Data Mining:

Data mining is the implementation of intelligent techniques for pattern extraction from data. These techniques involve the use of algorithms and statistical models to analyze large datasets and uncover hidden patterns, relationships, and insights. Data mining is widely used in various fields, such as marketing, finance, healthcare, and social media, to make informed decisions and drive business growth.

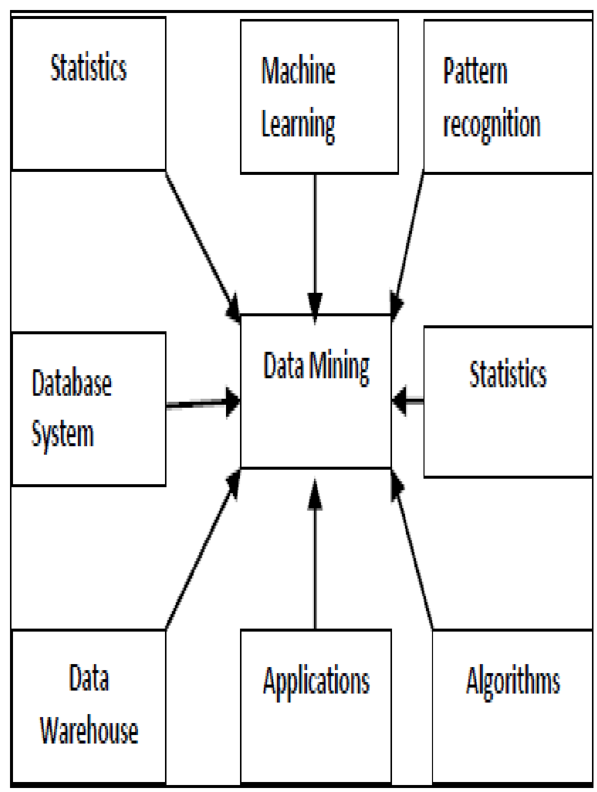

Figure. 1: Uses of data mining

Pattern Evaluation: The interest in the patterns produced by data mining is determined by the specific business problem at hand. Different businesses have different needs and objectives, which influence the level of interest in data mining patterns. For example, a retail company may be interested in identifying customer buying habits to improve marketing strategies, while a healthcare organization may focus on patterns in patient data to enhance treatment outcomes.

Knowledge Presentation: Knowledge Presentation aids the user in comprehending and interpreting the resultant patterns by employing visualization techniques to depict intriguing patterns.

While numerous other algorithms, such as Apriori and Eclat, are available for mining frequent item sets, the FP-Tree growth algorithm only performs two pre-processing steps: it performs an initial scan of the database to ascertain the item frequencies. As they are ineligible for frequent item sets, all uncommon items—those that do not occur in a minimum number of user-specified transactions—are eliminated. This pre-processing step helps to reduce the size of the database and improve the efficiency of the mining process. After eliminating uncommon items, the algorithm constructs an FP-Tree structure to represent the remaining transactions, which allows for efficient pattern mining.

1.3 FP Growth Algorithm

As an enhancement over the Apriori algorithm, the FP-Growth algorithm is a viable alternative for identifying the most frequently occurring frequent itemset in a given dataset. FP-Growth overcomes the shortcomings of Apriori by employing the tree-building technique to identify frequent item sets, thereby producing an algorithm that operates at a quicker rate. One notable characteristic of FP-Growth is its utilization of the FP-Tree, a tree-like data structure that enables the direct extraction of frequent item sets [33].

FP growth algorithm programming is an effective method for generating frequent item sets in the absence of prospective item set generation. The optimal tally is determined through the execution of two database searches utilizing a divide-and-conquer strategy. It is capable of extracting the products through the utilization of lift, leverage, and conviction, provided that a minimum threshold is set. [34]

FP Growth identifies the set of most frequent items without requiring candidate generation. It is divided into two stages: The FP-Tree, a compact data structure, is generated in step one, and the frequent item sets are extracted directly from the FP-Tree in step two.

. One benefit is its ability to generate a conditional pattern base from a database with minimal support requirements. Additionally, its compact structure and absence of candidate generation contribute to its reduced memory usage. A disadvantage is that it has difficulty processing vast pattern data collections.

The FP-Growth method comprises the subsequent three primary phases:

1- The initial phase involves the creation of a sub-database called the conditional pattern base, comprised of prefix paths and suffix patterns. This is accomplished by utilizing the FP-Tree that was previously constructed.
2- The FP-Tree conditional generation constitutes the second stage, during which the support count for each item in each conditional pattern base is consolidated. Afterwards, the conditional FP-Tree is employed to generate items whose support count exceeds the minimal support count.
3- Third and last, a frequent item-set search is performed. By combining items for each conditional FP tree, frequent item sets can be obtained if the conditional FP-Tree is a single path. When the conditional FP tree contains more than one path, recursion is used to generate the FP growth.

2. Analysis Of Exiting Algorithms

This segment provides an analysis and evaluation of the most recent and significant algorithms in this domain, contrasting their respective advantages and disadvantages. The analysis and evaluation of these algorithms will help in understanding their effectiveness and applicability in solving real-world problems. Additionally, it will shed light on any limitations or potential improvements that can be made to enhance their performance further. The following are components of these algorithms:

2.1 An efficient frequent pattern mining algorithm using a highly compressed prefix tree

An efficient frequent pattern mining algorithm utilizing a highly compressed prefix tree is concluded by the author in this paper. The recognition of recurring patterns is crucial in the process of mining association norms. A foundational algorithm for frequent pattern mining is FP-growth. To accomplish this, it utilizes a recursive mining process and a prefix tree structure (FP-Tree). On the contrary, the author argues that the performance of FP-growth is significantly influenced by the overall number of recursive calls, resulting in subpar performance when constructing multiple conditional FP-trees. This research was conducted utilizing a novel algorithm known as HCFP-growth. By data-mining a set of frequent item sets at a quicker rate than alternative algorithms, significant memory savings can be achieved in numerous scenarios [35]. HCFP-growth, the novel algorithm used in this research, outperforms alternative algorithms in terms of data mining for frequent item sets at a faster rate. This not only leads to significant memory savings but also proves to be advantageous in various scenarios. Therefore, HCFP-growth is a promising approach for improving the performance of FP-growth and constructing multiple conditional FP-trees efficiently [35].

The central proposal of the article is to develop a novel algorithm known as HCFP-growth. The implementation of this algorithm significantly diminishes the quantity of recursive invocations needed to extract complete frequent patterns. The purpose of this endeavor is to develop a highly compressed FP-tree (HCFP-tree). This results in an increase in prefix sharing and a decrease in the prefix tree's node count. The proposed methodology comprises three distinct stages. These stages include preprocessing, pattern growth, and post-processing. In the preprocessing stage, the dataset is transformed into a vertical format to improve efficiency. The pattern growth stage involves constructing the HCFP tree and extracting frequent patterns using the HCFP growth algorithm. Finally, in the post-processing stage, the extracted patterns are pruned and evaluated for their significance.

Step 1: FP-growth performs a double database scan. The initial survey identifies and counts the support for all frequent items. The algorithm then incorporates the frequently transacted items that were identified in the FP tree during the second scan.

Step 2: FP-growth generates a conditional pattern base and a conditional FP-tree for each item in the header table, after constructing the FP-tree. The aforementioned procedure is subsequently executed recursively on each conditional FP-tree until only one path remains in the conditional FP-tree. The FIs are subsequently obtained following the formation of every conceivable combination of nodes along the singular paths.

Step 3 involves calculating the FP-growth efficiency through the sum of all recursions. Reducing the number of recursive queries expedites the mining operation and eliminates the need to construct a greater number of conditional trees. Nevertheless, the quantity of recursive invocations is governed by the number of nodes in the tree and the degree of prefix sharing between patterns.

2.2 An Improved FP-Growth Algorithm Based on Projection Database Mining in Big Data

This paper provides a summary of the challenge encountered by the conventional FP-Growth frequent item-set mining algorithm and certain enhanced algorithms: the inability to independently store the enormous FP-tree in memory. The FP-Growth algorithm is utilized to construct the FP-tree through a sequence of predetermined operations: the transaction database D is initially searched to acquire all frequent 1-item sets; the frequent 1-item collection is then sorted in descending support count order. Subsequently, the algorithm constructs the corresponding conditional FP-tree for every frequent 1-item in the FP-tree and decomposes the frequent itemset within each conditional FP-tree. Ultimately, the collection of frequent items extracted from every conditional FP tree comprises every frequent item associated with the complete transaction database [36]. This process of constructing conditional FP-trees and decomposing frequent itemsets is repeated until no more frequent itemsets can be found. The resulting collection of frequent items provides valuable insights into the patterns and associations present in the transaction database.

The authors propose a parallel mining algorithm in this paper. Transaction databases are extracted based on the frequency of each item, and a projection database is generated for each frequent item. Subsequently, the projection databases are distributed to node machines, and the enhanced algorithm is executed on each node machine to generate partial frequent item sets via parallel mining. Finally, summarization is used to obtain all frequent item sets. This paper presents an algorithm that addresses the storage problem of isolated memory without requiring the generation of FP-trees for transaction databases, thanks to the utilization of a parallel algorithm. The proposed algorithm utilizes a parallel algorithm to address the storage problem of isolated memory without the need for generating FP-trees. This approach allows for efficient mining of partial frequent item sets on distributed node machines, resulting in improved scalability and performance. Additionally, the algorithm incorporates summarization techniques to obtain all frequent item sets from the generated partial sets.

Additionally, the efficacy of recursive mining of the element set is enhanced by this algorithm. The algorithm improves the efficiency of recursive mining by optimizing the process of searching through the element set. It eliminates redundant iterations and streamlines the recursive mining process, resulting in more effective and faster execution. Moreover, this enhancement enables the algorithm to handle larger element sets with improved performance.

The work steps in this paper are summarized as follows:

- **Step 1** involves the creation of a distinct database containing the transactions associated with each recurring item.
- **Step 2** involves the creation of an independent tree for each recurring item. During the creation of this tree, no nodes are added to the original item or any succeeding item sets. Instead, it is generated specifically for the item sets that come before this particular element.
- **Step 3:** The tree is reduced in size through the deletion of item sets that have a support count below 2.

2.3 Optimization of FP-Growth algorithm based on cloud computing and computer big data

The authors of this paper discuss the challenges encountered by frequent item-set mining algorithms when dealing with large datasets. They also highlight the deficiencies of frequent lists of elements, which are frequently cross-referenced during the construction of FP trees. The authors then attempted to establish a connection between the proliferation of outstanding cloud computing platforms and the exponential growth of cloud computing technology. These platforms for cloud computing offer an efficient method for handling large volumes of data. The purpose of this paper was to improve the inadequate extraction efficiency of the conventional FP-Growth algorithm in environments with large amounts of data by introducing an enhanced FP-Growth algorithm [37]. The enhanced FP-Growth algorithm proposed in this paper aims to address the limitations of the conventional FP-Growth algorithm, such as low extraction efficiency in environments with large amounts of data. By leveraging efficient platforms for cloud computing, the authors aimed to enhance the performance and scalability of the algorithm. Additionally, the authors conducted experiments to evaluate the effectiveness of their proposed algorithm and compared it with existing approaches in terms of extraction efficiency and computational resources utilized.

Table 1: Comparative study of various algorithms of frequent item set mining.

S.no	Search	Authors	Advantages	Drawbacks
1	An efficient frequent pattern mining algorithm using a highly compressed prefix tree	Zhu, at el	HCFP growth is among the quickest algorithms at all times. It also frequently uses the least amount of memory.	FP-tree construction in a database is difficult because it necessitates a sizable database.
2	An Improved FP-Growth Algorithm Based on Projection Database Mining in Big Data	Le Zhang at el	This algorithm eliminates the requirement for FP-tree generation in transaction databases and resolves the storage issue associated with autonomous memory.	The algorithm increases I/O burden to some degree by requiring the extraction of the transaction database for every frequent 1-item, the generation of a projection database corresponding to each frequent 1-item, and the distribution of the projection database to every node machine.
3	Optimization of FP-Growth algorithm based on cloud computing and computer big data	Zhang at el	A 13% reduction in traversal time and a 25% increase in mining efficacy are characteristics of the enhanced FP-growth algorithm, which outperforms the original algorithm. Furthermore, the implementation of this algorithm for data clustering optimizes performance, decreases error rates, and increases utility.	The algorithm's implementation and complexity present a degree of difficulty, rendering its introduction to the general public challenging. As an additional, This algorithm has stringent environmental requirements.

This article focuses primarily on FP growth algorithm optimization in the context of big data and cloud computing. The objective of this study was to develop a performance-enhancing algorithm that surpasses the original algorithm. Additionally, the authors sought to decrease the traversal time and enhance the mining efficiency. By employing this algorithm for data collection, the error rate can be reduced and performance can be improved, similar to the initial implementation of the two-dimensional support table method. The authors conducted experiments on a large dataset to validate the effectiveness of the optimized FP-Growth algorithm. The results showed significant improvements in both traversal time and mining efficiency, making it a promising solution for big data and cloud computing environments.

The content comprises the subsequent various components:

(1) Initially, you must generate a transaction set T, which provides a distinct view of the item contents and the order in which they appear for each transaction. Concurrently, this procedure generates the item header table and determines the frequency [38] by traversing the entire transaction set T and noting the number of occurrences of each item in each set.

(2) Following the acquisition of the frequency of each transaction item by traversing the transaction set T, the set is re-examined to generate the two-dimensional table of algorithm support count [39]. Begin by generating a two-dimensional support count table. Utilize the items in the item header table as the header and attribute header, respectively. The first column of the attribute header represents the content. Fill in the counter value obtained by traversing the transaction set into the corresponding table's box, by the attributes of the support count two-dimensional table. As an illustration, if the item of transaction T2 is denoted by (I1, I4), increment the count in column I4 and row I1 of the data set; if the item of transaction T4 is denoted by (I1, I2, I4), then add 1 to the two-dimensional vector table containing items (I1, I2), (I1, I4), and (I2, I4) [40].

Thirdly, the two-dimensional support count table represents the occurrence frequency of each pairwise combination of items in the transaction set. As the algorithm disregards the support count and the greater of the set value and the set value, certain deletions must be performed once the table has been constructed successfully [41]. The support count list reveals that items I4 and I5 fail to satisfy the minimum support item requirements; therefore, they are explicitly excluded from the algorithm's scope. However, the support count for the aforementioned three items remains in the two-dimensional table; remove the rows and columns.

3. Conclusion

This research paper examines contemporary algorithms utilized in the extraction of frequent item sets. This analysis outlines the fundamental concepts and procedures of every algorithm. This research paper conducts an exhaustive analysis of several algorithms designed to identify frequent patterns. Additionally, it explains how these algorithms can be utilized to extract frequent patterns from diverse datasets. Although the first algorithm has the benefit of requiring little memory, constructing a tree from a large database is not a simple task. The second algorithm does, to some degree, increase the I/O burden. In addition to having the capability to enhance mining efficiency, the third algorithm possesses a degree of implementation and complexity difficulty. Compared to the other algorithms, a novel model is suggested that will require less time to execute, handle large volumes of data, and have a low error rate.

4. References

[1] Jamsheela, O., & Raju, G. (2023). An improved frequent pattern tree: the child structured frequent pattern tree CSFP-tree. Pattern Analysis and Applications, 26(2),437-454, https://link.springer.com/article/10.1007/s10044-022-01111-1

[2] GPM, S., NKH, N., Shah, M. A., & SKD, K. (2022). FOEH: Frequent Pattern Mining Performance Optimization over Large Transactional Data in Extended Hadoop MapReduce,1-20, https://europepmc.org/article/ppr/ppr577613

[3] Gan, W., Lin, J. C. W., Fournier-Viger, P., Chao, H. C., Hong, T. P., & Fujita, H. (2018). A survey of incremental high-utility itemset mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(2), e1242, https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1242

[4] Senthilkumar, A., & Hariprasad, D. (2021). A Spark Based Frequent Itemset Mining Using Resource Management for Implementation of Fp-Growth Algorithm in Cloud Environment. Annals of the Romanian Society for Cell Biology, 6050-6059, http://www.annalsofrscb.ro/index.php/journal/article/view/3170

[5] Wang, L., Meng, J., Xu, P., & Peng, K. (2018). Mining temporal association rules with frequent itemsets tree. Applied Soft Computing, 62, 817-829., https://www.sciencedirect.com/science/article/abs/pii/S1568494617305525

[6] Memar, M., Deypir, M., Sadreddini, M. H., & Fakhrahmad, S. M. (2012). An efficient frequent itemset mining method over high-speed data streams. The Computer Journal, 55(11), 1357-1366, https://ieeexplore.ieee.org/abstract/document/8130915

[7] Chang, J. R., Chen, Y. S., Lin, C. K., & Cheng, M. F. (2021). Advanced data mining of SSD quality based on FP-growth data analysis. Applied Sciences, 11(4), 1715, https://www.mdpi.com/2076-3417/11/4/1715

[8] Zhang, C., & Teng, G. (2021). Application of an improved association rule algorithm in rural development assessment in China. In IOP Conference Series: Earth and Environmental Science (Vol. 772, No. 1, p. 012089). IOP Publishing, https://iopscience.iop.org/article/10.1088/1755-1315/772/1/012089/meta

[9] Lin, Y., Lin, F., Huang, D., & Fang, X. (2021). Voltage sag severity analysis based on improved FP-Growth algorithm and AHP algorithm. In Journal of Physics: Conference Series (Vol. 1732, No. 1, p. 012088). IOP Publishing., https://iopscience.iop.org/article/10.1088/1742-6596/1732/1/012088/meta

[10] Li, Y., & Yin, S. (2020). Mining algorithm for weighted FP-tree frequent item sets based on two-dimensional table. In Journal of Physics: Conference Series (Vol. 1453, No. 1, p. 012002). IOP Publishing.,

https://iopscience.iop.org/article/10.1088/1742-6596/1453/1/012002/meta
[11] Zhang, B. (2021). Optimization of FP-growth algorithm based on cloud computing and computer big data. International Journal of System Assurance Engineering and Management, 12(4), 853-863, https://link.springer.com/article/10.1007/s13198-021-01139-2
[12] Srikant, R., & Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In Proceedings of the 1996 ACM SIGMOD international conference on Management of data (pp. 1-12), https://dl.acm.org/doi/abs/10.1145/233269.233311
[13] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499), https://www.it.uu.se/edu/course/homepage/infoutv/ht08/vldb94_rj.pdf
[14] Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). New algorithms for fast discovery of association rules. In KDD (Vol. 97, pp. 283-286), https://cdn.aaai.org/KDD/1997/KDD97-060.pdf
[15] Burdick, D., Calimlim, M., Flannick, J., Gehrke, J., & Yiu, T. (2005). MAFIA: A maximal frequent itemset algorithm. IEEE transactions on knowledge and data engineering, 17(11), 1490-1504, https://ieeexplore.ieee.org/abstract/document/1512035
[16] Zheng, J., Zhang, D., Leung, S. C., & Zhou, X. (2010). An efficient algorithm for frequent itemsets in data mining. In 2010 7th International Conference on Service Systems and Service Management (pp. 1-6). IEEE., https://ieeexplore.ieee.org/abstract/document/5530116
[17] Deng, Z. H., & Lv, S. L. (2014). Fast mining frequent itemsets using Nodesets. Expert Systems with Applications, 41(10), 4505-4512, https://www.sciencedirect.com/science/article/abs/pii/S0957417414000463
[18] Vo, B., Coenen, F., Le, T., & Hong, T. P. (2013). A hybrid approach for mining frequent itemsets. In 2013 IEEE International Conference on Systems, Man, and Cybernetics (pp. 4647-4651). IEEE, https://ieeexplore.ieee.org/abstract/document/6722546
[19] Wang, J., Han, J., & Pei, J. (2003). CLOSET+ searching for the best strategies for mining frequent closed itemsets. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 236-245), https://dl.acm.org/doi/abs/10.1145/956750.956779
[20] Song, M., & Rajasekaran, S. (2006). A transaction mapping algorithm for frequent itemsets mining. IEEE transactions on knowledge and data engineering, 18(4), 472-481, https://ieeexplore.ieee.org/abstract/document/1599386
[21] Wang, J., Han, J., Lu, Y., & Tzvetkov, P. (2005). TFP: An efficient algorithm for mining top-k frequent closed itemsets. IEEE Transactions on knowledge and data engineering, 17(5), 652-663, https://ieeexplore.ieee.org/abstract/document/1411744
[22] Woon, Y. K., Ng, W. K., & Lim, E. P. (2004). A support-ordered trie for fast frequent itemset discovery. IEEE Transactions on Knowledge and Data Engineering, 16(7), 875-879, https://ieeexplore.ieee.org/abstract/document/1318569/
[23] rAO, d. s. (2021). a survey on frequent item set mining for large transactional data. information technology in industry, 9(2), 885-893, http://it-in-industry.org/index.php/itii/article/view/426
[24] Woon, Y. K., Ng, W. K., & Das, A. (2001). Fast online dynamic association rule mining. In Proceedings of the Second International Conference on Web Information Systems Engineering (Vol. 1, pp. 278-287). IEEE, https://ieeexplore.ieee.org/abstract/document/996489/
[25] Ji, L., Zhang, B., & Li, J. (2006). A new improvement on apriori algorithm. In 2006 International Conference on Computational Intelligence and Security (Vol. 1, pp. 840-844). IEEE, https://ieeexplore.ieee.org/abstract/document/4072208/
[26] Deng, Z., & Wang, Z. (2010). A new fast vertical method for mining frequent patterns. International Journal of Computational Intelligence Systems, 3(6), 733-744, https://www.tandfonline.com/doi/abs/10.1080/18756891.2010.9727736
[27] Deng, Z., Wang, Z., & Jiang, J. (2012). A new algorithm for fast mining frequent itemsets using N-lists. Science China Information Sciences, 55, 2008-2030, https://link.springer.com/article/10.1007/s11432-012-4638-z

[28] Vijayalakshmi, V., & Pethalakshmi, A. (2013). Mining of Frequent Itemsets with an Enhanced Apriori Algorithm. International Journal of Computer Applications, 81(4), https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0e3ab50aef4e9e2e31bded4f324351263a239ef6

[29] Abd Elhady, S. Z., Ghali, N. I., Abo-Elfetoh, A. F. A. F., & Idrees, A. M. (2022). Exploratory Big Data Statistical Analysis The Impact Of People Life's Characteristics On Their Educational Level. Journal Of Theoretical And Applied Information Technology, 100(5), http://www.jatit.org/volumes/Vol100No5/25Vol100No5.pdf.

[30] Mi, X. (2022). The Mining Algorithm of Maximum Frequent Itemsets Based on Frequent Pattern Tree. Computational Intelligence and Neuroscience, 2022, https://www.hindawi.com/journals/cin/2022/7022168/

[31] Alsaeedi, H. A., & Alhegami, A. S. (2022). An Incremental Interesting Maximal Frequent Itemset Mining Based on FP-Growth Algorithm. Complexity, 2022, https://www.hindawi.com/journals/complexity/2022/1942517/

[32] Novita, R., & Salisah, F. N. (2021). Determination of the relationship pattern of association topic on Al-Qur'an using FP-Growth Algorithms. In IOP Conference Series: Materials Science and Engineering (Vol. 1088, No. 1, p. 012020). IOP Publishing, https://iopscience.iop.org/article/10.1088/1757-899X/1088/1/012020/meta

[33] Lael, T. A. D., & Pramudito, D. A. (2023). Use of Data Mining for The Analysis of Consumer Purchase Patterns with The Fpgrowth Algorithm on Motor Spare Part Sales Transactions Data. IAIC Transactions on Sustainable Digital Innovation (ITSDI), 4(2), 128-136,https://aptikom-journal.id/itsdi/article/view/582

[34] Patil, M., & Patil, T. (2022). Apriori Algorithm against Fp Growth Algorithm: A Comparative Study of Data Mining Algorithms. Available at SSRN 4113695,https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4113695

[35] Zhu, X., & Liu, Y. (2019). An efficient frequent pattern mining algorithm using a highly compressed prefix tree. Intelligent Data Analysis, 23(S1),153-173, https://content.iospress.com/articles/intelligent-data-analysis/ida192645

[36] Le Zhang, S. X., Li, X., Wu, X., & Chang, P. C. (2019). An Improved FP-Growth Algorithm Based on Projection Database Mining in Big Data.,https://bit.nkust.edu.tw/~jihmsp/2019/vol1/JIH-MSP-2019-01-10.pdf

[37] Zhang, B. (2021). Optimization of FP-growth algorithm based on cloud computing and computer big data. International Journal of System Assurance Engineering and Management, 12(4),853-863,https://link.springer.com/article/10.1007/s13198-021-01139-2

[38] Su, H. T., Cheung, S. H., & Lo, E. Y. M. (2018). Multi-objective optimal design for flood risk management with resilience objectives. Stochastic environmental research and risk assessment, 32, 1147-1162, https://link.springer.com/article/10.1007/s00477-017-1508-7

[39] Ebrahim, M. A., Elyan, T., Wadie, F., & Abd-Allah, M. A. (2017). Optimal design of RC snubber circuit for mitigating transient overvoltage on VCB via hybrid FFT/Wavelet Genetic approach. Electric Power Systems Research, 143,451-461, https://www.sciencedirect.com/science/article/pii/S037877961630476X

[40] Deng, C., Zhou, Y., Jiang, W., & Feng, X. (2017). Optimal design of inter-plant hydrogen network with purification reuse/recycle. International Journal of Hydrogen Energy, 42(31), 19984-20002, https://www.sciencedirect.com/science/article/pii/S0360319917326307

[41] Wang, H., Zhao, Y., Ma, X., & Wang, H. (2017). Optimal design of constant-stress accelerated degradation tests using the M-optimality criterion. Reliability Engineering & System Safety, 164, 45-54., https://www.sciencedirect.com/science/article/pii/S095183201630285X