EasyChair Preprint

№ 12251

A Systematic Review of Frequent Itemset Mining Algorithms and Their Capabilities

Anshu Singla and Parul Gandhi

February 24, 2024

A Systematic Review of Frequent Itemset Mining Algorithms and Their Capabilities

Anshu Singla, Research scholar Manav Rachna International Institute of Research and Studies (MRIIRS)
Faculty of Computer Application
Faridabad,Haryana
anshu008in@gmail.com

Dr. Parul Gandhi, SCA Professor,
Manav Rachna International Institute of Research and Studies (MRIIRS),
Faculty of Computer Application
Faridabad,Haryana
Parul.sca@mriu.edu.in

ABSTRACT:

FIM (frequent itemset mining) is a key data analysis task. that is important for large businesses decision-making. The efficient mining of frequent patterns from transactional databases has been proposed using a variety of FIM methods. This work gives a thorough evaluation of numerous conventional and parallel FIM methods, outlining the benefits and drawbacks of each.The classic Apriori algorithm creates candidate itemsets using breadth-wise searching. Due to the numerous database searches, it has a significant memory and calculation time consumption .On the other hand, FP-Growth compresses the whole database into an FP-Tree after just one scan, but creating the FP-Tree can take some time for large databases.Modern algorithms like MMFI and Max-IFP use FP-Arrays and FP-Matrix to get over these restrictions, which leads to more effective memory usage and quicker frequent pattern generation. When compared to FP-Growth, the COFI algorithm uses less memory since it takes a pruning strategy. Enhanced variations, including Eclat and CP-Miner, introduced strategies to address memory consumption and runtime efficiency. Additionally, emerging paradigms like parallel and distributed FIM algorithms have gained prominence with the advent of big data. However, challenges remain in terms of scalability, noise handling, and pattern diversity. This comparison study makes it clear that a more potent and scalable FIM algorithm is therefore essential. This systematic literature review aims to provide a comprehensive analysis of various FIM algorithms, highlighting their strengths, limitations, and applications across different domains. Future research will concentrate on improving the suggested method and running comprehensive trials to assess its effectiveness and scalability.

I. INTRODUCTION:

Data mining is a method for extracting pertinent information from a vast number of datasets. These patterns are advantageous for making decisions, cutting costs, generating income, and fostering market competitiveness. Although it seems straightforward, accessing important and necessary information from the database is not possible. Data mining is a complex procedure made up of several simple procedures. With the increasing development in measurement, it appears to be relatively easy to think about mining.

A. Association rule mining:

Association rule mining is a key approach in the field of data mining, aimed at uncovering hidden relationships and patterns within large datasets. It enables us to identify associations among items, attributes, or events that frequently co-occur. These associations are expressed in the form of "if-then" rules, where the presence of certain items or attributes implies the likelihood of the presence of others. This technique has found applications in a wide range of domains, from retail and market analysis to healthcare, recommendation systems, and beyond. In retail, for example, association rule mining allows businesses to understand customers' purchasing behavior and preferences. By identifying items frequently purchased together, retailers can optimize their product placement, design targeted promotions, and enhance overall customer satisfaction. Association rule mining involves two primary concepts: frequent itemsets and association rules. Collections of items that are very frequently appear together are called frequent itemsets. in transactions above a specified support threshold. Support indicates the frequency of occuring of an itemset in the database. Association rules, on the other hand, express the relationships between itemsets. These rules are derived from the frequent itemsets and are typically evaluated based on two

metrics: confidence and lift. When the antecedent item(s) are present, confidence quantifies the possibility that the subsequent item(s) will also exist., Lift assesses the link between the antecedent and subsequent items' strength beyond what would be predicted by chance. association rule mining remains a powerful tool for data exploration, decision-making, and pattern discovery. Its applications continue to expand as new algorithms and techniques address scalability concerns and provide more meaningful and actionable insights from complex datasets.

B. **Freqent Itemsets:**

Itemsets with occurrences that are at least that threshold number are commonly seen in a dataset. Frequent itemsets are a group of itemsets that frequently appear in data sets. Frequent itemset mining scans a sizable transactional database dataset for recurrent itemset sequences. The relationship between these itemsets is then established, resulting in useful association rules.

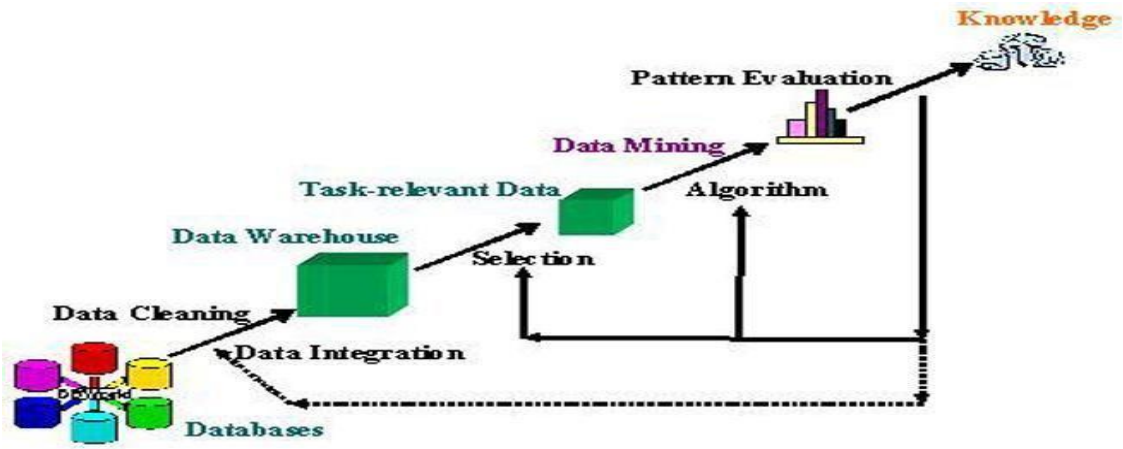

Figure 1: Knowledge Discovery and Data Mining (Source: Han and Kamber (2001)

Although many academics have suggested several FIM algorithms to improve the procedure, further development is still required to increase the efficiency of FIM algorithms. FIM algorithms have two main drawbacks: first, they take a long time to execute or get more complex over time, and second, they consume a lot of memory when building large datasets to extract frequent item sets. This is because these algorithms cannot handle the enormous dataset, which is constantly growing. The aim of this research is to create a powerful FIM algorithm capable of extracting all pertinent patterns from a continuously expanding data set. The main purpose of this research is to comprehend the pros and cons of the most well-known itemset mining techniques. Frequent itemset mining methods are used to present a new and powerful FIM method. In this study, we give a comprehensive review of significant frequent itemset mining techniques and some of their extensions and adjustments.The literature on apriori algorithms and expansions, the FP- Growth method and its variations, and parallel frequent itemset mining was evaluated after a general overview of data mining and frequent item mining. Part 2 displays the thorough literature assessment of FIM algorithms. Part 3 presents the findings and recommendations from the comparison of FIM methodologies. The final section of this essay concludes its analysis and makes recommendations for additional research.

II. LITERATURE REVIEW

FIM algorithms have evolved significantly, from the foundational Apriori to the efficient FP-Growth and specialized algorithms like CP-Miner. They continue to be an essential tool for uncovering hidden patterns in diverse applications. As data continues to play a pivotal role in decision-making, FIM algorithms are poised to remain a crucial element in the data mining landscape. A systematic literature review of FIM algorithms is discussed as below.

A. **Apriori Algorithm**

Agrawal and colleagues (1994) gave the first conventional algorithm the name Apriori. Using a huge transactional database, the AprioriHybrid algorithm, which he described, mines association rules. Due to the two algorithms that make up this algorithm's combination, it enables database and transaction size scale-up possibilities. Agrawal find closure feature in frequent k-itemsets. These k-itemset will only have frequent subitemsets if it has frequent k-itemsets as well. Frequent itemsets 1 are produced from a database scan. Itemsets.1 are now being utilised to generate candidates for itemsets-2, and this process will carry on this till k-itemsets can be processed from itemsets-1.a result of the efficiency of the as apriori approach and the production of multiple huge candidate itemsets. The usefulness of the traditional apriori method proposed by Xiao et al. The ideas of support optimisation (support>1/2) and confidence-based item removal served as the foundation for this technique. To reduce database scanning and candidate key creation, this revised version employs tagged transaction compression. This enhancement increases capacity while reducing complexity to around 80% of the original apriori method.Aditya et al. claim that the apriori technique has a number of flaws because it doesn't work with real data, which frequently changes in databases. Every time the in apriori technique is used, the database must be searched for potential candidate keys, which makes it more difficult to produce more itemsets rapidly and necessitates more data storage space. The author of this study suggests a space- and time-complexity-increasing algorithm that generates common patterns using real data. Due to not reading the database for every iteration, the total number of combinations is decreased from 2N to 2N-2.

B. **Frequent Growth Algortithms:**

More frequent patterns must be generated since the FP-tree must be explored more slowly. The author used an FP-array and an FP-Matrix to introduce the unique optimised algorithm MMFI. The size of the FP-tree is decreased using the FP-array and matrix, which decreases the time needed to traverse a large FP-tree. Results versus the traditional Fp-Max algorithm show how effective this method is in terms of space and time. O. Jamsheela (2015) focused on the significance of The frequent pattern growth approach was significantly improved in the beginning by Han et al. (2000). Multiple database scanning is costly for the apriori algorithm, hence a list of candidate keys is generated., and a lack of storage capacity for these candidate key creation. But some of the database searching was diminished. A novel FP-growth approach has been created to produce frequent itemsets without employing candidate construction of itemsets. This method improves performance in terms of scalability and works well in both time and space. Hui Ling and colleagues (2012) concentrate on extracting the most frequent patterns through the use of array and matrix generation. The cornerstone of the FPGrowth methodology is the divide and conquer tactic. FPGrowth simply needs to conduct one database search in order to build a frequent items list. These item sets are derived in decreasing order by itemset frequency. This itemset's frequency list is used to construct an FP-tree that represents the full database. Frequent patterns are now constructed utilising conditional FP trees rather than the entire database. By Grahne et al. (2005), the performance of the FP-tree-based technique is improved. The entire database is compressed into a single FP- tree after performing a preliminary database scan. In FP-Growth algorithms, frequent itemsets must be derived by traversing the FPtree. Granhe offers a novel method called FP arrays in place of FP trees. It is unnecessary to navigate an FP-tree while using an FP-array. This approach works well with collections of sparse data. Grahne also published methods for closed and maximal frequent item sets. FP-tree and FP-array are used with optimisation methods to create efficient algorithms. The approach was time and space efficient for normal dense data, but this wasn't viable for sparse datasets. The FP-tree has so far worked effectively for frequent itemsets list. Wang et al. (2008) present a novel way to enhance the performance of an existing algorithm. The recommended algorithm is called L-FP _tree. The first part of this technique involves scanning the database and building equivalence classes from each itemset. It is necessary to first remove unnecessary and superfluous item sets according to equivalence classes before building potentially

optimal Level FP- tree. TAN advances the 2009 workdone from Grahne by The FP-tree is built in the database's initial scan and conditional FP-tree is generated essential concepts required to construct an FP-Tree.
The FP-tree outperforms the apriori approach in terms of performance. The FP-Tree algorithm as well as other non-apriori algorithms are modified by Jamsheela. Numerous publications have already discussed the utility of the FP-Tree. Zhaopang et al. (2018) found that when the itemset is large, the FP-Tree branch extends farther. In that case, storing this branch would demand a larger amount of space, increasing the complexity of the FIM method. The author suggested the New FP-Tree as a remedy for this issue because it can reduce the amount of space needed for large itemsets. The efficiency of the Max-IFP technique was also provided by the author for mining the most frequent patterns. Results from experiments show that FP-Tree augmentation produces better outcomes.

C. **Pruning algorithms:**

Pruning algorithms play a crucial role in improving efficacy and effectiveness of mining association rules by reducing the search space and eliminating irrelevant or redundant itemsets. This literature review provides an overview of existing research on pruning algorithms in association rule mining, highlighting their contributions, advantages, and limitations.

- **FP-Growth with Pruning:**

The FP-Growth algorithm is a popular pruning-based approach that introduces conditional FP- trees and path-based pruning. By avoiding the generation of candidate itemsets, FP-Growth significantly reduces memory consumption and improves mining efficiency. This algorithm has become widely adopted due to its ability to efficiently mine frequent itemsets without the need for costly candidate generation.

- **COFI-Tree Pruning:**

The COFI algorithm is an optimization technique to mine frequently used itemsets from high-speed transactional database. COFI employs a pruning-based strategy to remove infrequent itemsets from the FP-Tree, leading to reduced memory consumption and faster mining. This approach is particularly useful for handling large-scale data streams where memory efficiency is crucial.

- **Pruning Techniques for Efficient Association Rule Mining:**

This study investigates pruning techniques for efficiently mining multiple minimum supports and association rules. The proposed approach optimizes the mining process by reducing computational overhead and improving the standard of association rules mined. The algorithm may concentrate on producing rules with different levels of significance thanks to the usage of various minimal supports. Pruning Redundant Rules in Frequent Itemset Mining: This research proposes efficient pruning methods to eliminate redundant rules generated during frequent itemset mining. By removing redundant rules, the algorithm improves the interpretability and usefulness of the mined association rules and reduces the computational overhead associated with rule generation.

- **Top-K Pruning for Focused Mining:**

The CAEP algorithm introduces a top-K pruning technique to focus on the most significant and interesting itemsets based on user-defined metrics. By prioritizing the most relevant itemsets, CAEP enhances the usefulness and interpretability of the mined association rules, making it valuable in specific applications with a targeted search space.Pruning algorithms have played a crucial role in association rule mining, significantly improving the efficiency and quality of the mining process. Studies have introduced various pruning techniques, such as path-based pruning, redundant rule elimination, and top-K pruning, to optimize frequent itemset generation and enhance the interpretability of association rules.

D. **Parallel Frequent Itemset Mining:**

Wang (2003) created the Par-MinMax parallel frequent pattern mining algorithm to mine maximal frequent patterns including parallelization techniques such as task parallelism, data parallelism, and hybrid approaches.Backtrack pruning techniques are employed to increase algorithm performance by reducing input output overload. To get around the FP-Growth mining technique's lengthy execution time, Javed et al. (2004) presented the parallel FP-Growth algorithm, also known as PFP (parallel FP Growth). PFP algorithm distributes the entire workload evenly across all nodes, much like FP-tree data structure. PFP utilises a master-slave paradigm. For its database, each CPU creates a LOCAL HEADER TABLE (LHT). Master node creates the GHT (global header table). In order to control these slave nodes, GHT sends information to them. PFP is a better performing parallel frequent mining technique due to this information exchange property, Wang (2003) developed the algorithm in parallel, Par-MinMax by decreasing input output overload, backtrack pruning techniques are used to improve algorithm performance. The parallel FP-Growth algorithm, also known as PFP, was introduced by Javed et al. (2004) to circumvent the FP-Growth mining technique's excessive execution time (parallel FP Growth). PFP method, like FP-tree data structure, equally distributes the full burden among all nodes. The master-slave paradigm is used in PFP. Each CPU produces a LOCAL HEADER TABLE for its database (LHT). The GHT is made by the master node (global header table). GHT communicates with these slave nodes and controls them. Due to this information exchange property, PFP performs better than frequent mining in parallel.Tree-partition technique was utilised by Chen (2006) .Then divide this FP-Tree into sections to share work across the threads. The task was parallelized using a dynamic scheduling heuristic technique. The load balance issue is solved using dynamic scheduling. To demonstrate the effectiveness of the suggested approach for comparison with known parallel methods and experimental results, many types of datasets were used. Scalability was made possible by adding more CPUs. As dataset size rose, algorithm performance improved.

In order to solve the load balancing problem, Yu et al. (2007) introduced a parallel solution for in this study. LFP is contrasted with PFP. Yu and colleagues (2012) developed the MapReduce parallel method BIDE-MR. The Apache Hadoop environment was used to construct the algorithm. A significant contribution using the ECLAT algorithm in the MapReduce architecture is made by Moens et al. (2013). There are two approaches. The first is Dist-ECLAT, which divides search space rather than data space. The second method, BigFM, adopts an apriori approach for mining k-size itemsets before switching to ECLAT. It employs a mixed strategy.Yan et al. presented PCAFP, a limited parallel frequent mining approach in a MapReduce setting (2015). Frequent patterns that are confined are those that depend on restrictions imposed by customers in order to increase mining productivity.In their 2016 publication, Pei, B., Wang, X, and Wang F. proposed the PNFP algorithm, which uses the existential probability of itemsets in transactions to establish association rules.

Using the MapReduce architecture, PNFP is a parallelized algorithm that is more efficient. Due to its parallelization and use of the MapReduce framework to handle large datasets, PNFP is more effective. It reduces the cost of data replication and communication mining patterns frequently. a number of the several MapReduce framework-based techniques proposed by Chang (2017), additional work is still required to boost scalability due to the substantial workload.

For huge data, Shaik and colleagues (2018) developed a parallel algorithm. While working on the parallel algorithm, the author concentrates on three elements: load balancing, work segmentation, and memory scalability. The efficiency and expandability of memory of the proposed approach are tested using a variety of datasets, including those from the Mashroom, Chess, Census, and Connect.

Hadoop technology, as revealed by Dawen Xiahow (2018), offers superior results than spatial and temporal Frequent Patterns using MapReduce.The MR-PFP algorithm is a parallel frequent pattern growth technique suggested in this study. MapReduce is replaced with Hadoop technology for parallelization.The findings demonstrate that the MR-PFP method improves efficiency and scalability over the older Parallel FP-growth (PFP) approach. A quick and scalable FIM technique was suggested by Chon and Kim (2018) using the MapReduce platform. This approach is beneficial since it has minimal network connection overhead, no intermediate data issues, and no workload skewness issues.Using the antimonotone property, Luna, Padillo, Luna, Herrera, and

Ventura (2018) devised the FIM algorithms. We studied various Map-Reduce exhaustive FIM methods for big data.Currently, Map Reduce-based (non-exhaustive) FIM techniques are in the development stage.
Martn et. alMRQAR .'s is the first non-exhaustive MapReduce algorithm they introduced in 2018. In 2018, Padillo et al. proposed the non-exhaustive method G3P-LSCusing the MapReduce infrastructure and the genetic programming language for routine pattern mining.In order to achieve high efficiency using multi item support frequent pattern, Whang (2019) developed a novel MISFP (multi item support frequent pattern) data analytic paradigm in hadoop-implemented parallel algorithms. According to experimental results, utilising the MISFP algorithm in a distributed setting reduces the temporal complexity of a parallel method by 38%.
Al-Bana(2022) introduces SHFIM, a novel Spark-based Hybrid Frequent Itemset Mining algorithm that leverages the strengths of both traditional Apriori-like methods and advanced Spark-based parallel processing. This algorithm performed Integration with apache Spark for parallel processing and scalability with efficient distribution of data and computations across a cluster of nodes. Pre-post and Pre-post+ are the least expensive options for the Accidents dataset. It features an innovative N-list data structure called PPC-tree.

III. DICUSSION

This section summarizes the review study of FIM algorithms. Table 1 describes Key Observations from FIM Algorithms of the fundamental frequent itemset mining algorithms. After reviewing a a richness of literature, it can be concluded that existing algorithms cannot handle the large amounts of data that are being generated constantly. As frequent patterns helps with decision-making to develop new steps for large enterprises in today environment, frequent itemset mining is a particularly focused subject. Therefore, a more noticeable, effective algorithm is required .

Table 1: Key Observations from FIM Algorithms

S.No	Algorithms	Referneces	Findings
1.	Apriori algorithm	Agrawal & Shrikant (1994)	This algorithm use breadth-based searches using k-itemsets to find k+1 item sets. As the database is scanned more than once to provide candidate items, there is a significant calculation time and memory consumption.
2.	FP-Growth	Han, Pei & Yin, (2000)	This algorithm reduces all of the data into a single fp-tree after a single database scan. Fp-tree can be used for FIM instead of repeatedly using the entire database. The consumption memory and construction of fp tree of is large.
3.	MMFI algorithm	Zhaopang & Pan (2018)	In this algorithm ,fp-array and fp-matrix are used instead of fp-tree. fp-tree is more efficient and consumes less memory.
4.	Max-IFP algorithm	Wang, Feng, Zhang & Liao (2014) and C. Wang, Lin & Chang (2017)	For this frequent mining algorithm, an IFP tree is used by keeping important, frequent Patterns. The minimum support as criteria is really low.
5.	COFI algorithm	El-Hajj,& Zaiane O. R. (2003)	Memory consumption is minimised in comparison to fp-growth since the pruning approach is employed to create the COFI-tree. COFI algorithm performance declines in sparse databases.

IV. CONCLUSION AND FUTURE WORK

This systematic review thoroughly explored various FIM algorithms and their capabilities, shedding light on their strengths, weaknesses, and potential applications across diverse domains. The study emphasized the significance of efficient pattern mining in supporting business decisions, cost-cutting strategies, revenue generation, and competitive market positioning. The review encapsulated the evolution of FIM algorithms, starting from foundational methods like the Apriori algorithm and progressing to more optimized approaches such as FP-Growth. Modern innovations, such as the MMFI algorithm and Max-IFP, employ FP-Arrays and FP-Matrix to overcome limitations, resulting in more efficient memory usage and quicker pattern generation. Pruning-based algorithms like COFI mitigate memory consumption, and parallel/distributed FIM algorithms have emerged to handle big data. However, challenges persist in terms of scalability, noise handling, and pattern diversity. We have very briefly summarized this review in table1. Even additional development is used to reduce the compilation time and storage usage of FIM methods. The study's primary goal is to create an effective algorithm for frequent item mining.The ongoing evolution of FIM algorithms is essential to enable organizations to extract valuable insights from data and make informed, strategic decisions.

REFERENCES

[1] Sikharam, Ushamanjari & Sabnis, Vikrant & Jain, Jay & sikharam, Usha manjari. (2023). A REVIEW PAPER ON FREQUENT PATTERN MINING ON BIG DATA BY USING MAP REDUCE PROGRAMMING MODEL.

[2] Sikharam, Ushamanjari & Sabnis, Vikrant & Jain, Jay & sikharam, Usha manjari. (2023). A REVIEW PAPER ON FREQUENT PATTERN MINING ON BIG DATA BY USING MAP REDUCE PROGRAMMING MODEL.

[3] Ding, S.; Li, Z.; Zhang, K.; Mao, F. A Comparative Study of Frequent Pattern Mining with Trajectory Data. Sensors 2022, 22, 7608.

[4] S. Samiei, M. Joodaki and N. Ghadiri, "A Scalable Pattern Mining Method Using Apache Spark Platform," *2021 7th International Conference on Web Research (ICWR)*, Tehran, Iran, 2021, pp. 114-118, doi: 10.1109/ICWR51868.2021.9443111.

[5] El-Haji & Zaiane (2003). Co-occurrence frequent itemsets: An optimization technique for mining frequent itemsets from high-speed transactional data streams. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 45-54).

[6] Ca, Luo & Han(2003). Efficient and effective pruning methods for mining a large number of association rules. In Proceedings of the 3rd IEEE International Conference on Data Mining (pp. 21-28).

[7] Dong, Li. & Wong (1999). CAEP: A new efficient algorithm for mining frequent itemsets. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 398-402).

[8] Chee, Chin-Hoong(2018) ―Algorithms for Frequent Itemset Mining: a Literature Review.‖ ArtificialIntelligence Review, vol. 52, no. 4, 2018, pp. 2603–2621.

[9] Using Modified Anti-Monotone Support 2018‖ 2018 IEEE 10th International Conference on Humanoid,Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)

[10] Anastasiu, David -Big Data Frequent Pattern Mining ‖ Frequent Pattern Mining, 2014, pp. 225–259

[11] Caroro&Roseclaremath-An Enhanced Frequent Pattern-Growth Algorithm with Dual Pruning Using Modified Anti-Monotone Support, 2018 IEEE 10th International Conference on Humanoid

[12] Agrawal, R., Imieliński, T., & Swami, A. (1993). "Mining association rules between sets of items in large databases." ACM SIGMOD Record, 22(2), 207-216.

[13] Han, J., Pei, J., & Yin, Y. (2000). "Mining frequent patterns without candidate generation." In ACM SIGMOD Record, 29(2), 1-12.

[14] Zaki, M. J. (2000). "Scalable algorithms for association mining." IEEE Transactions on Knowledge and Data Engineering, 12(3), 372-390.

[15] Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). "New algorithms for fast discovery of association rules." In KDD'97 Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, 283-286.

[16] Srikant, R., & Agrawal, R. (1996). "Mining quantitative association rules in large relational tables." In Proceedings of the ACM SIGMOD International Conference on Management of Data, 1-12.

[17] Toivonen, H. (1996). "Sampling large databases for association rules." In VLDB '96: Proceedings of the 22nd International Conference on Very Large Data Bases, 134-145.

[18] Park, J. S., Chen, M. S., & Yu, P. S. (1995). "An effective hash-based algorithm for mining association rules." In Proceedings of the ACM SIGMOD International Conference on Management of Data, 175-186.

[19] Borgelt, C., & Kruse, R. (2002). "Induction of association rules: Apriori implementation." Competence Center for Applied Industrial Mathematics, Preprint 04/02.

[20] Borgelt, C. (2005). "An implementation of the FP-growth algorithm." In Proceedings of the 1st international workshop on open source data mining: Frequent pattern mining implementations, 1-5.

[21] Kavitha, V., & Kalaiselvi, K. (2011). "A survey of algorithms for frequent pattern mining in data mining." International Journal of Computer Applications, 17(6), 1-5.*