Chapter 4

Graph Based Itemset Mining

4.1 Motivation for Graph Based Data Algorithm

Data are ubiquitous and the quantity keeps on multiplying daily in many domains such as bioinformatics, information sciences, social networks, and stocks revealing the more complex and structural relationship between data items. Such data requires processing and storage capacity. The data is further used for visualizing and analyzing for knowledge extraction. When the data grows at a tremendous rate, there is a struggle to extract the knowledge. Moreover, most of the data generated is unstructured and unorganized. A method is required for extraction of knowledge and predicting the future [5]. Furthermore, to speed up the process of data mining in large databases, there is a need to develop a more compact way of representing the itemsets. A method to uncover the hidden patterns is to identify the common substructures in the data. The researchers have developed various techniques to unveil the relationships in the database to fathom the problem. One such solution is graph mining. Graph mining has gained importance as a result of its application in numerous domains. To understand and analyze such complex relationships, a graph-based algorithm helps to simplify and generalize the relationships. It has shown better results concerning time complexities. Therefore, a graph-based algorithm is more efficient when handling large data sets.

4.2 Introduction to Graph Based Data Structure

Graph-based representation is one of the most prominent tools used for the representation of data in terms of nodes [56]. Minimizing the input and the expenses involved in the operation is one of the targets of the mining algorithm. Using graph mining techniques, the performance of the mining algorithm is highly enhanced due to its ability to handle large datasets[21]. A graph-based approach is also used for detecting outliers. The experimental results show that a graph-based approach can handle massive data[27].

Graph mining is a technique for analyzing the data and creating a graph [30]. The graph approach is used for discovering and predicting extrinsic behavior. It also helps in minimizing computational resources and reduces the repetition of extracted features.
A graph consists of a set of {V, E} where V is a finite set of vertices and E consists of elements of edges. A vertex can also be expressed as a node and an edge is represented as a link. The vertex can be an element and the link represents the connection to any vertex. Edge can be binary where '1' is when it is connected and '0' is when it is not connected. It can contain some additional information like weighted based on directed or non-directed. Similarly, a node can simply be a point or it can contain additional detail like size [59]. There are two types of graphs: directed and undirected. An undirected graph consists of edges represented as $\{u,v\}$ denoting an undirected edge between u and v. A directed graph consists of ordered edges represented as (u,v) denoting a directed edge from vertex u to v. The vertices are linked together by an edge. The task of mining and analysis is easy when the data is represented using graph-based representation. The data is represented in terms of nodes and edges of the graph [56].
Graphs can be represented by two data structures: adjacency matrices and adjacency lists. When the level is high, the data structures use an array that is indexed by vertices. A unique integer is assigned as an identifier for each vertex ranging from 1 to V. In other words, these vertices are the integers [46].

4.3 Proposed Graph Based Algorithm

Numerous mining algorithms incorporated graphs for the process of mining. Most of the existing algorithms cannot address large datasets. The proposed itemset representation is incorporated in the proposed graph for mining the dataset. The graph G=$\{V, E\}$ where a vertex is expressed as an itemset and an edge contains links from one itemset to another one. The proposed graph-based algorithm consists of:

- The proposed itemset is incorporated in the graph-based algorithm.
- The graph-based algorithm uses the approach of Apriori for mining the itemsets.
- In the proposed graph-based algorithm, each itemset is mapped to a vertex on the graph.
- The associations of the subset and superset of the itemset are maintained using a graph-based data structure.

The steps of the proposed graph-based algorithms are as follows:

- The dataset is scanned to find the itemsets.
- The itemsets are then mapped to a graph where each itemset represents a node in the graph.
- An itemset is added to the graph only when its support count is equal to or greater than the threshold value.
- The itemset is maintained at each level based on the size of the candidate.
- The itemsets of the same cardinality are self-joined to generate the next candidate itemsets. The support count of these itemsets is compared with the threshold value.
- If the support count is less than the threshold value then these itemsets are pruned from the graph. This process continues until no itemset is left.

The flowchart of the proposed itemset is given in Figure 4-1.

4.4 Experimental Results

The rule mining algorithms are applied to the five datasets. They are as follows:

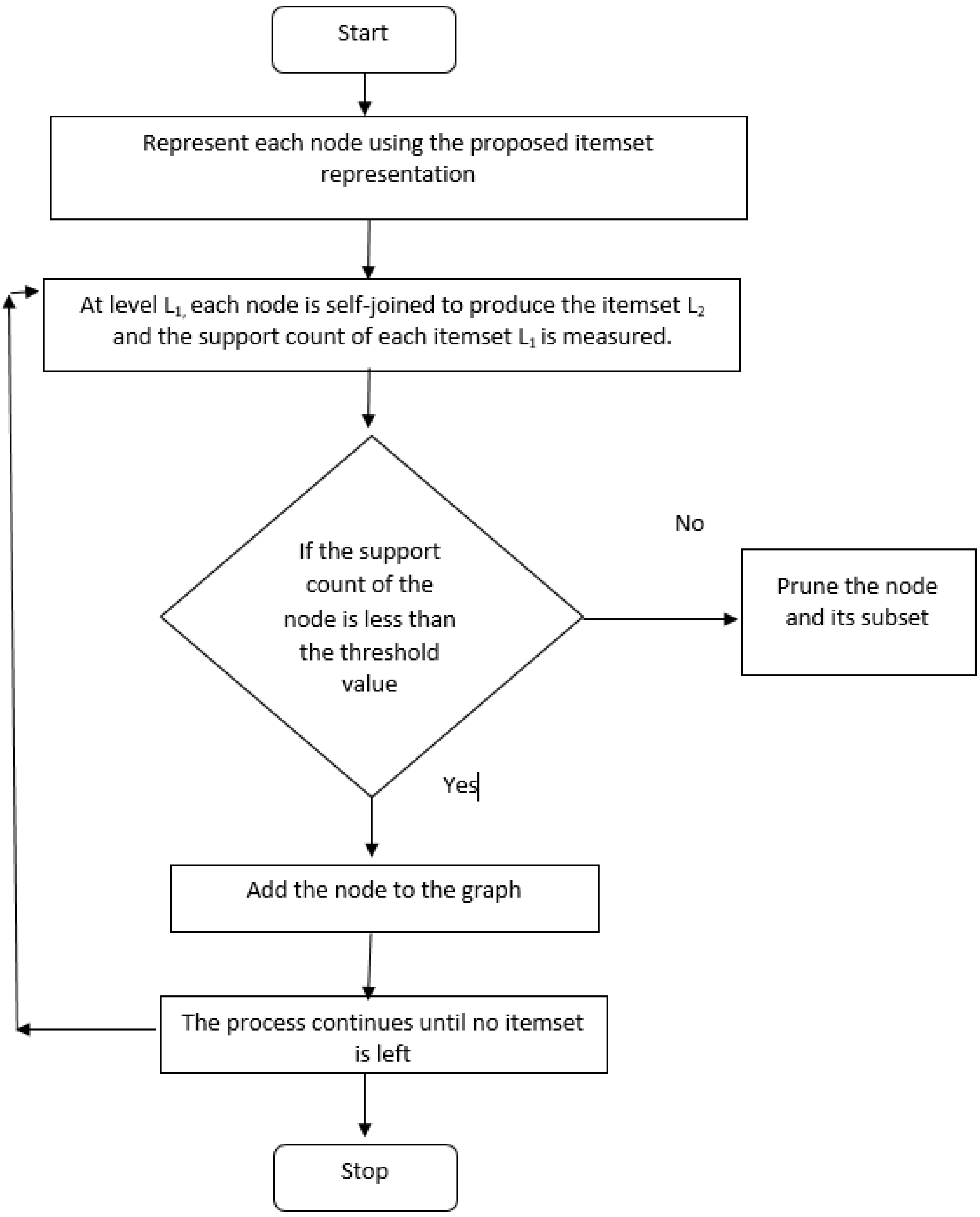
Start
Represent each node using the proposed itemset representation
At level L_1, each node is self-joined to produce the itemset L_2 and the support count of each itemset L_1 is measured.
If the support count of the node is less than the threshold value
No
Prune the node and its subset
Yes
Add the node to the graph
The process continues until no itemset is left
Stop

Figure 4-1: Flowchart of the proposed graph-based algorithm

- Car Evaluation Database: the dataset[1] contains structural information relating to car consisting of 1728 instances and 6 attributes.
- Bitcoin Heist Dataset: The dataset [2] contains features on a Bitcoin network for identifying ransomware payment consisting of 2916697 instances.
- Spatial Dataset: The dataset[3] was build by adding information of elevation to a 2D road network in Denmark. The dataset consists of 434874 instances.
- Hydraulic Dataset: The dataset[4] consist of the condition assessing of a hydraulic test.It collects the data from the sensor on the hydraulic rig consisting of 2205 instances and 43680 attributes.
- Cancer Dataset: This dataset[5] consist of 3D CT scans of humans. The dataset consists of 200000 attributes and 20000 instances.

The experiment is performed on CPU 2.30Ghz. The rule mining algorithms are executed for the different variations of the support count of 1%, 2.5 %, 5% and 10%. The datasets are applied to the proposed graph-based algorithm, Apriori Algorithm, and FP tree Algorithm. The execution time and memory consumption for the three algorithms are shown in Figure 4-2, Figure 4-3, Figure 4-4, Figure 4-5, Figure 4-6, Figure 4-7, Figure 4-8, Figure 4-9, Figure 4-10, Figure 4-11.

4.5 Evaluation of the Proposed Algorithm

The performance of the proposed graph-based algorithm is analyzed on datasets based on the distinct test. Both the real and sparse datasets having different characteristics are considered. The datasets were acquired from the UCI repository. For the experimental settings, a 64-bit, 4 GB RAM, and Core i5 are used for experimenting. The frequent itemset is produced based on the evaluation metric, support count. During the experiments, no other processes were running in the background. The datasets along with their execution time and memory usage are shown in Table 4.1.
As observed from Table 4.1, the time and memory consumption for a different dataset with different support counts varies. The behavior of the mining algorithms changes according to the characteristics and size of the dataset. When

[1]https://archive.ics.uci.edu/ml/datasets/Car+Evaluation
[2]https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddressDataset
[3]https://archive.ics.uci.edu/ml/datasets/3D+Road+Network+(North+Jutland,+Denmark)
[4]https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems
[5]https://archive.ics.uci.edu/ml/datasets/Deepfakes%3A+Medical+Image+Tamper+Detection

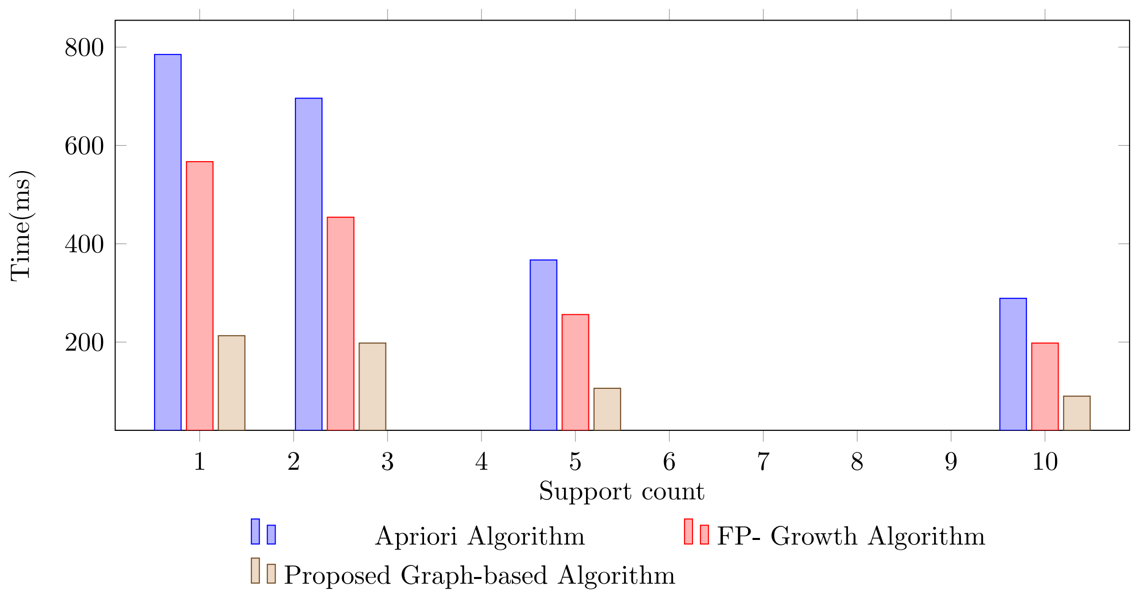

Figure 4-2: Time(millisecond) consumption of Different Rule Mining Algorithms for Car dataset of support count of 1%, 2.5 %, 5% and 10%

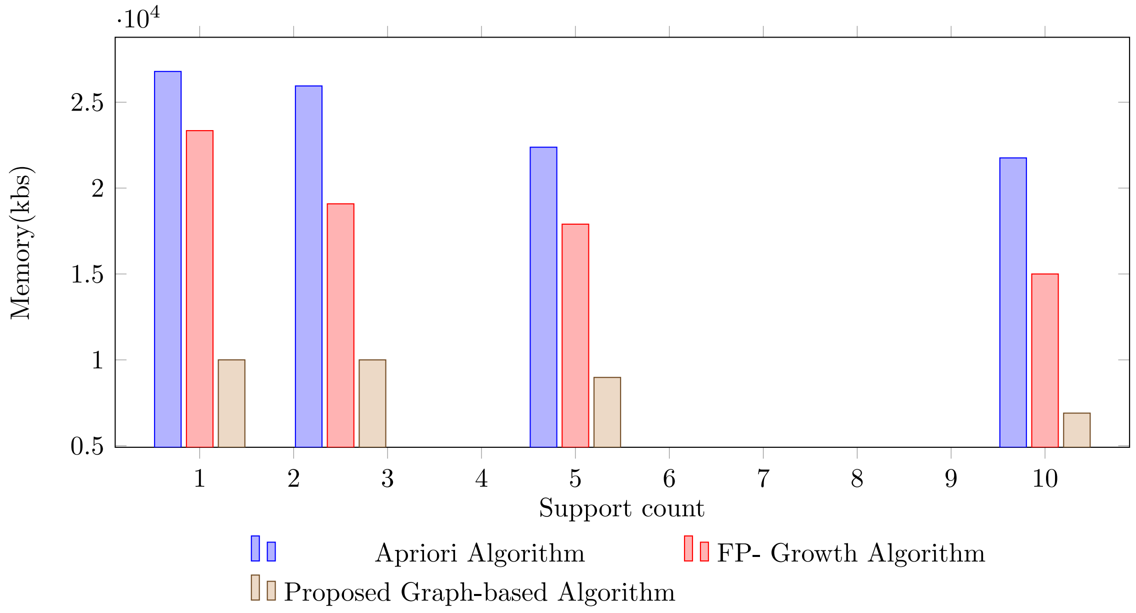

Figure 4-3: Memory(kilobits) consumption of Rule Mining Algorithms for Car dataset of support count of 1%, 2.5 %, 5% and 10%

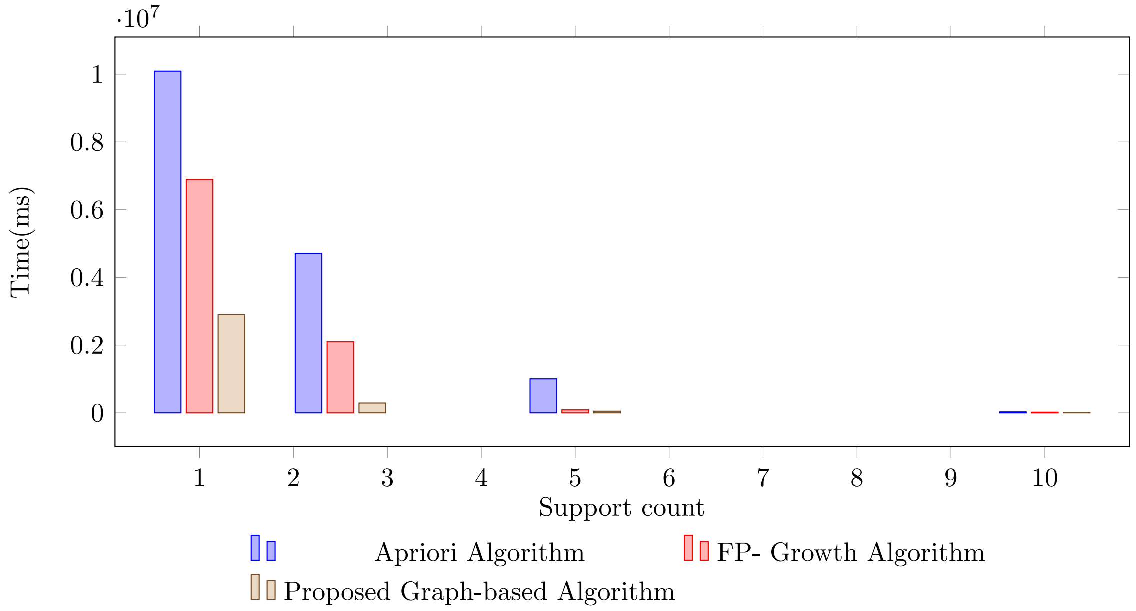

Figure 4-4: Time(millisecond) consumption of Different Mining Algorithms for Bitcoin Heist dataset of support count of 1%, 2.5 %, 5% and 10%

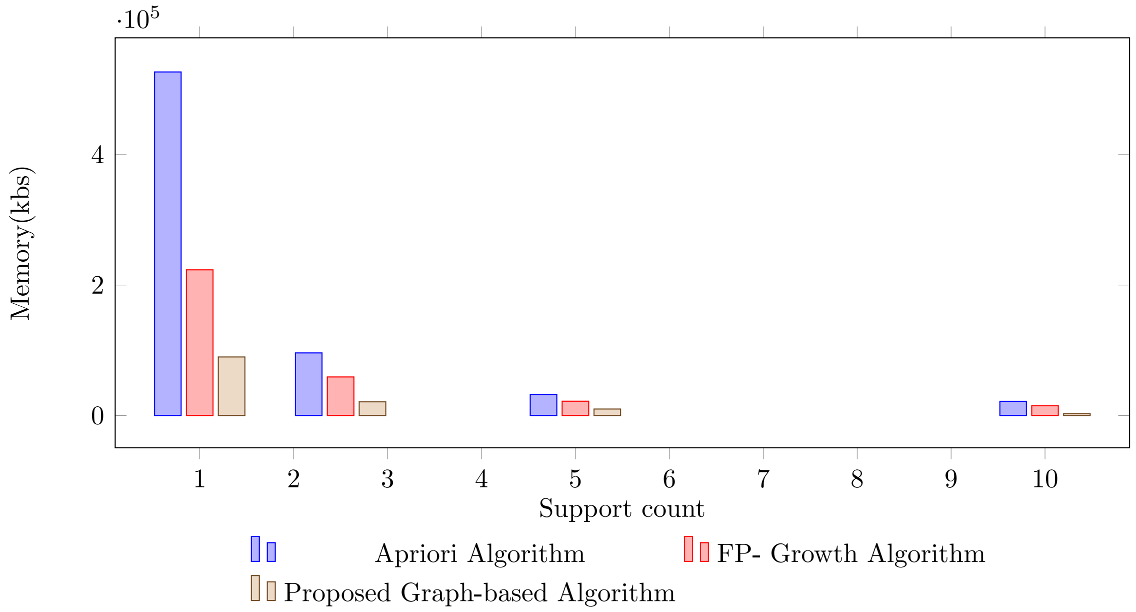

Figure 4-5: Memory(kilobits) consumption of Different Mining Algorithms for Bitcoin Heist dataset of support count of 1%, 2.5 %, 5% and 10%

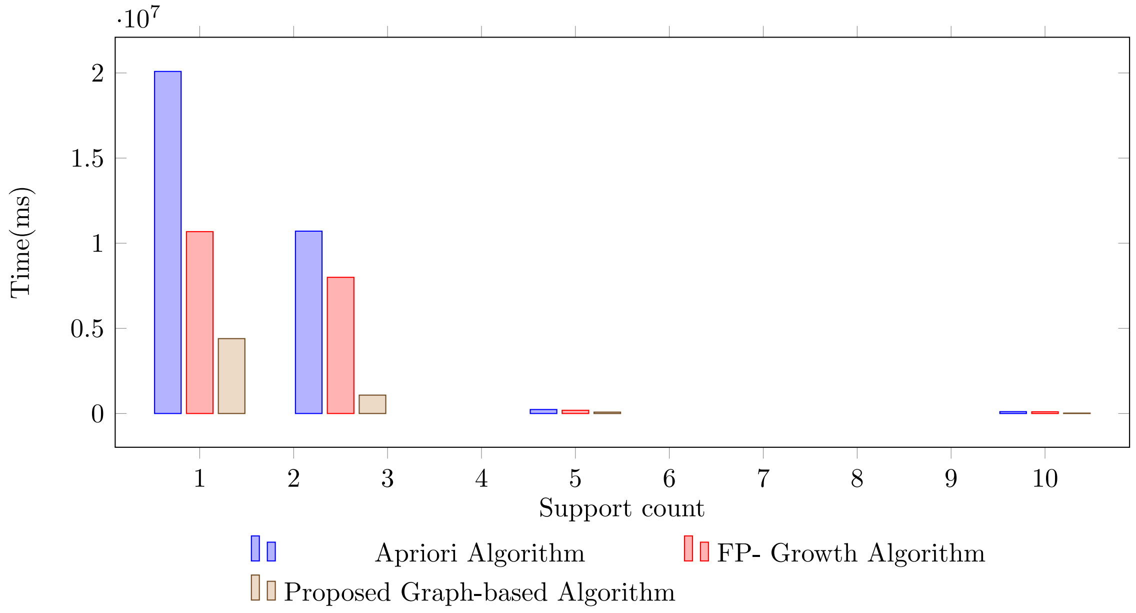

Figure 4-6: Time(millisecond) consumption of Different Mining Algorithms for Spatial dataset of support count of 1%, 2.5 %, 5% and 10%

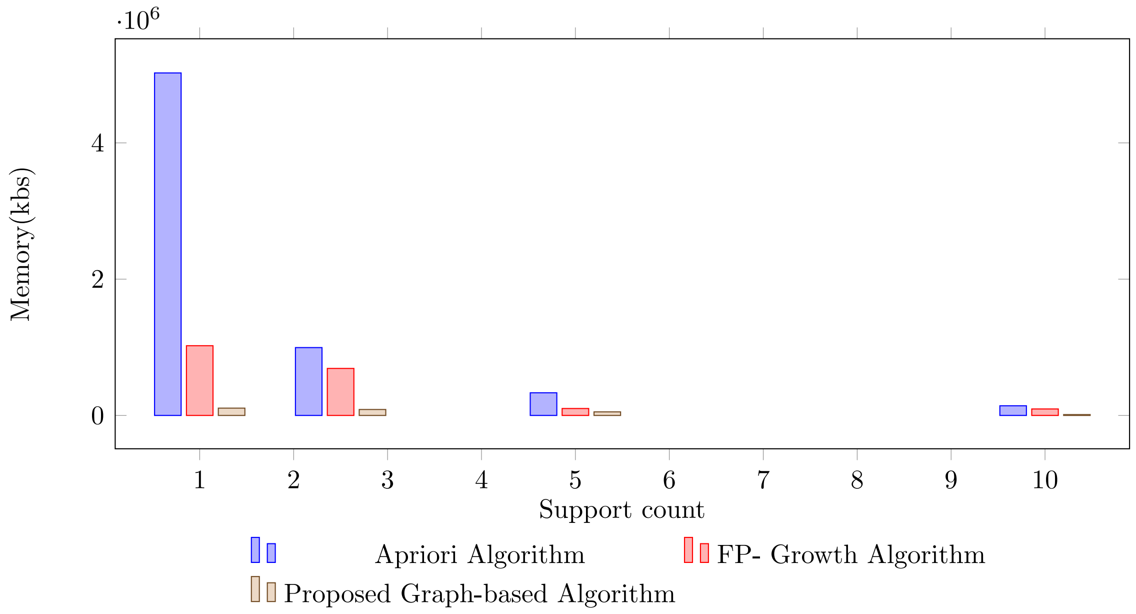

Figure 4-7: Memory(kilobits) consumption of Rule Mining Algorithms for Spatial dataset of support count of 1%, 2.5 %, 5% and 10%

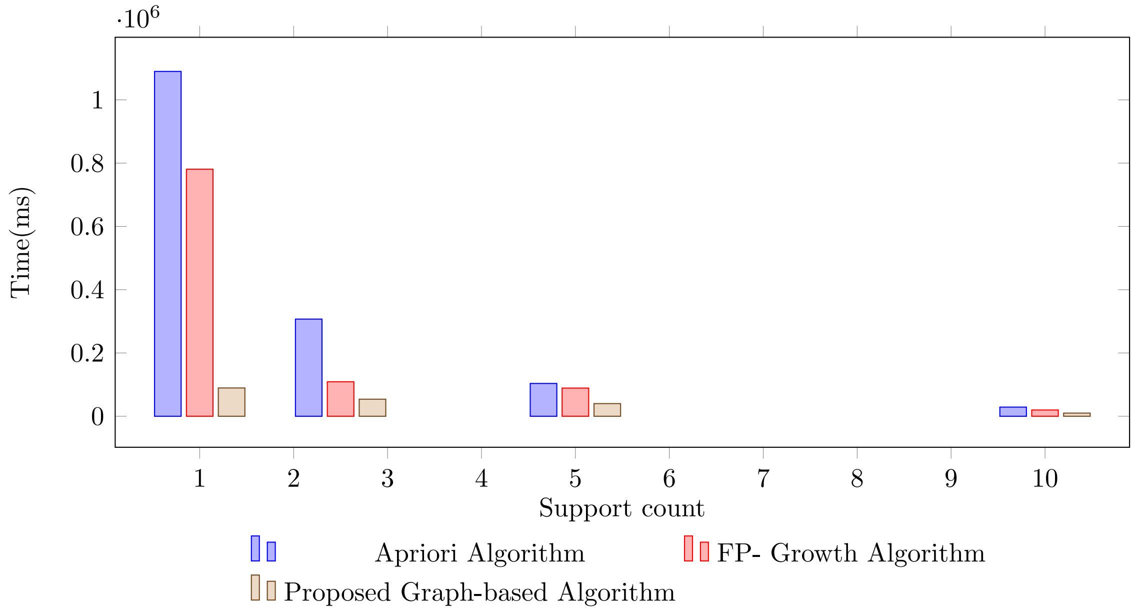

Figure 4-8: Time(ms) consumption of Rule Mining Algorithms for Hydraulic dataset of support count of 1%, 2.5 %, 5% and 10%

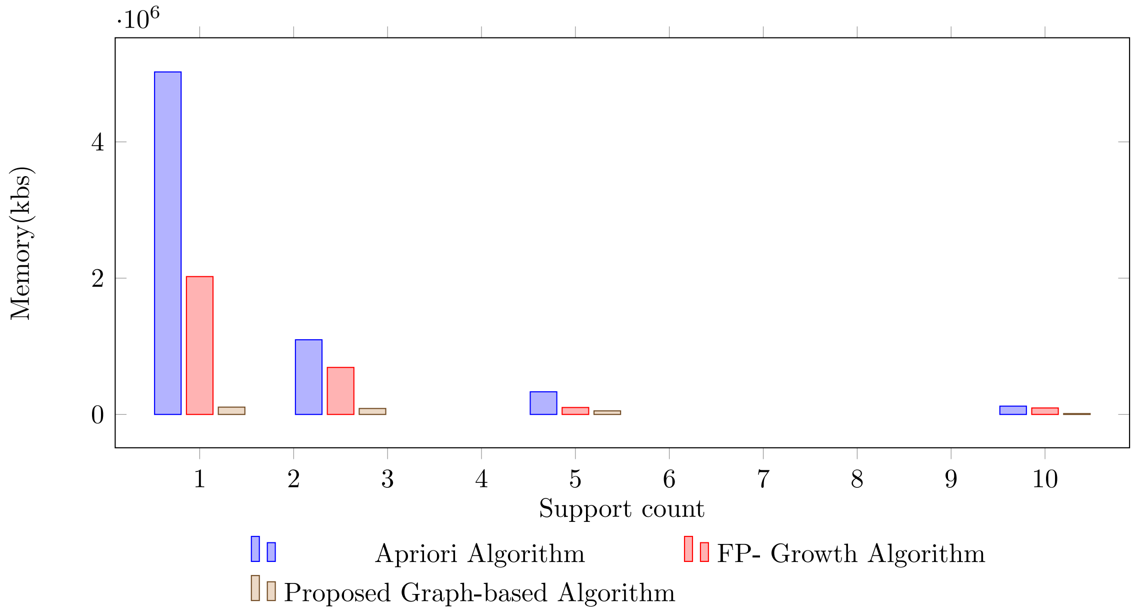

Figure 4-9: Memory(kilobits) consumption of Different Mining Algorithms for Hydraulic dataset of support count of 1%, 2.5 %, 5% and 10%

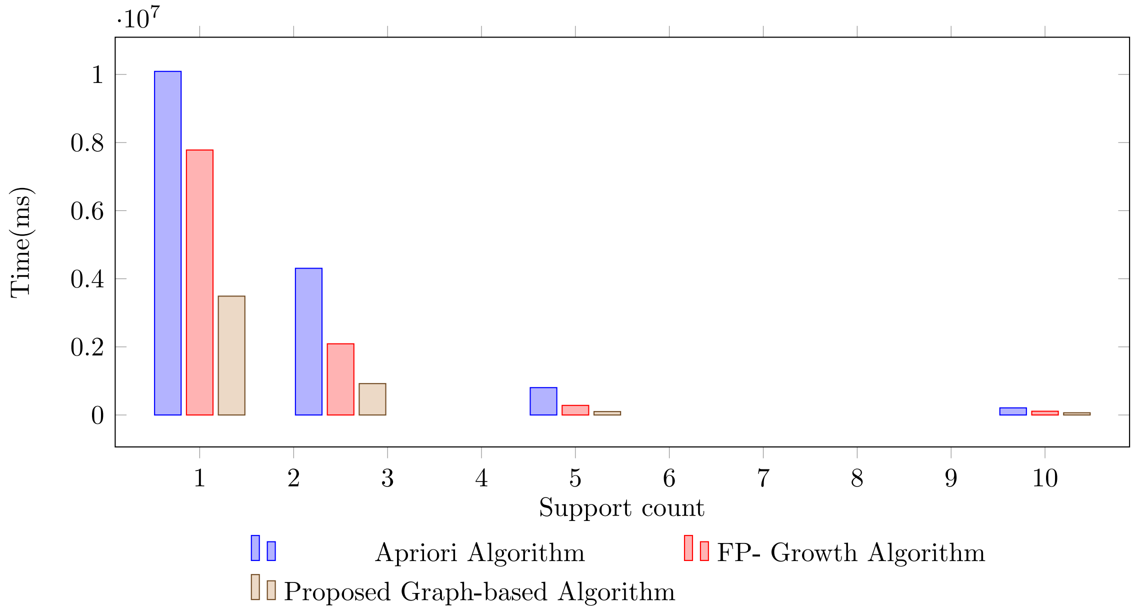

Figure 4-10: Time(millisecond) consumption of Rule Mining Algorithms for Cancer dataset of support count of 1%, 2.5 %, 5% and 10%

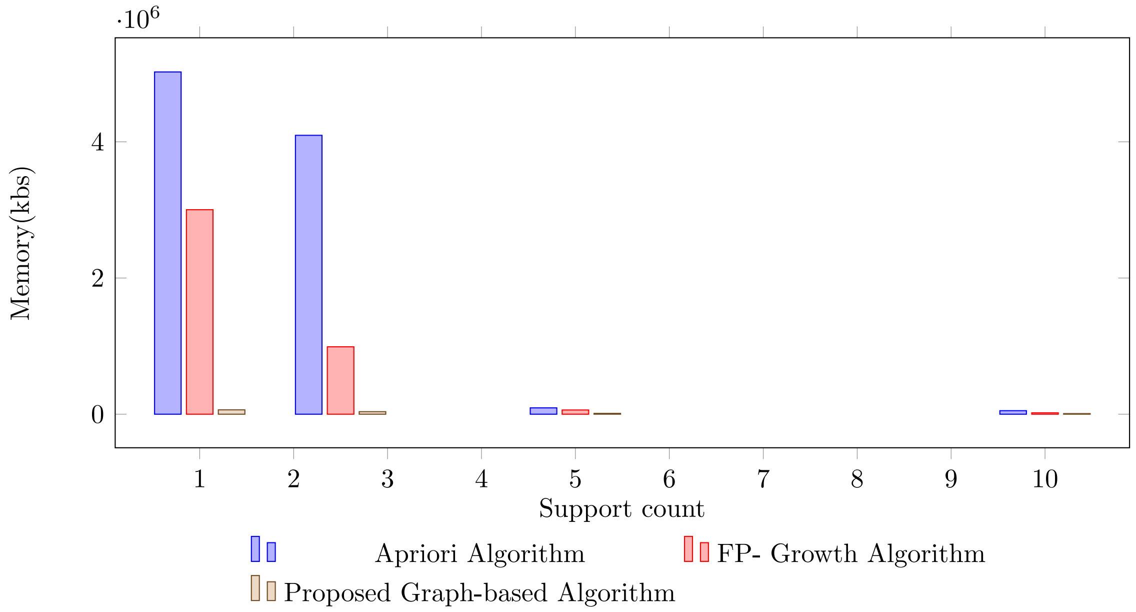

Figure 4-11: Memory(kilobits) consumption of Rule Mining Algorithms for Cancer disease dataset of support count of 1%, 2.5 %, 5% and 10%

Table 4.1: Time(millisecond) and Memory(kilobits) consumption using Apriori, FP-Tree, and Proposed graph-based algorithm for the different datasets

Sl.No	Dataset	Support	Apriori		FP-Tree		Proposed graph	
			Time (ms)	Memory (kbs)	Time (ms)	Memory (kbs)	Time (ms)	Memory (kbs)
1.	Car	1%	785	26790	567	23345	213	10000
		2.5%	696	25990	454	19084	198	9999
		5%	367	22380	256	17923	106	8978
		10%	289	21758	193	15000	90	6900
2.	Bitcoin Heist	1%	10089111	526790	6890800	224411	2897967	89729
		2.5%	4709696	95998	2099045	59084	289899	20929
		5%	1003679	32380	89090	21900	48975	9900
		10%	28909	21758	19809	15000	9089	2900
3.	Spatial	1%	20089122	5096554	10699876	1029099	4390000	107272
		2.5%	10709696	995947	7999065	690084	1079491	92424
		5%	233679	332380	199032	101900	81489	55890
		10%	108909	141758	99866	95000	23467	12398
4.	Hydraulic	1%	1077990	4996992	7790819	1003345	8945l6	103989
		2.5%	307077	1095947	109045	678903	53988	87421
		5%	103559	345665	89122	101987	40001	52398
		10%	28984	121779	19867	97867	9991	10397
5.	Cancer	1%	10089990	5987889	7680889	3003345	3489449	64272
		2.5%	4308990	4095947	2090405	990084	921921	37428
		5%	803679	93238	281090	61900	100001	14598
		10%	207789	51758	108827	20500	64976	8979

the support count is minimized, the time and memory consumption is maximized and when the support count is maximized, then the time execution and memory consumption is minimized. This implies that when the support count is less, the number of generated frequent itemsets is more. In addition, the processing time and computation, the mining operation involving read and write operation also increases the execution time and memory consumption.

4.6 Proof of Correctness

The objective is to show that the itemsets maintained by the Graph-based algorithm Tree are the same as those of the Apriori Algorithm.

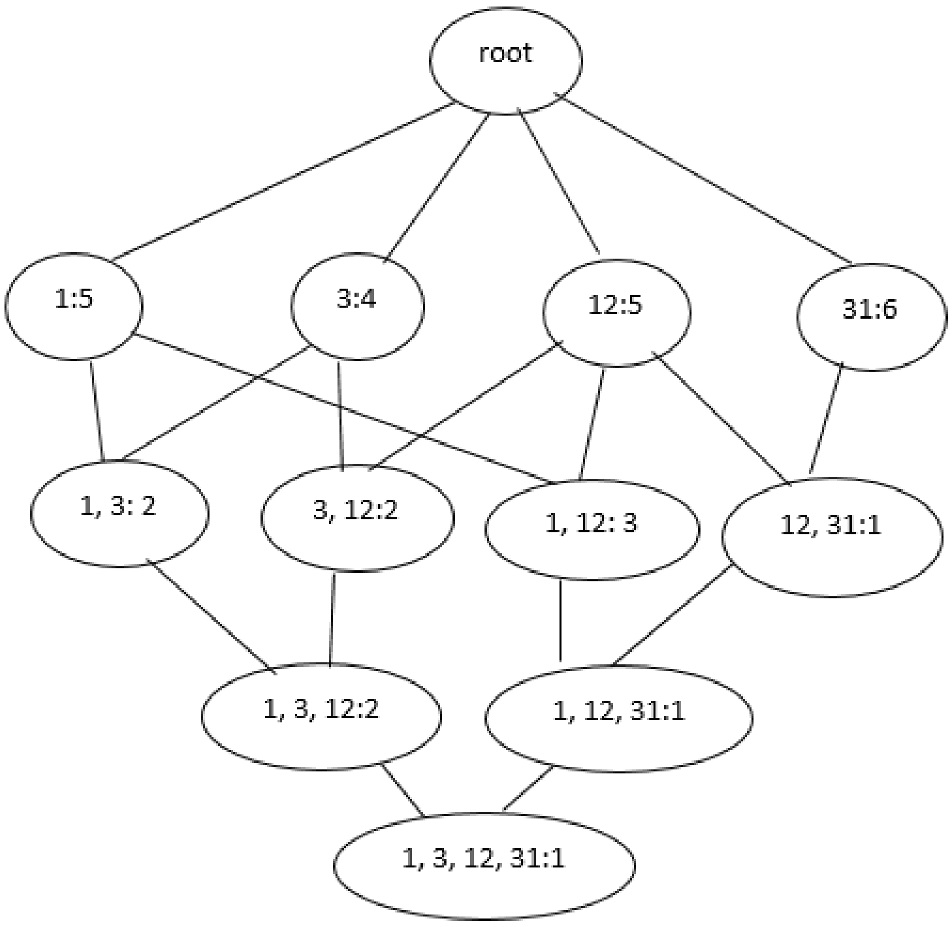

Figure 4-12: An example of the proposed graph-based algorithm

Lemma 1.1 Given a set of items, the frequent itemset generated from the dataset after representing the itemset using the proposed itemset representation, the procedure produces the same frequent itemsets produced by Apriori.

Proof The itemset is represented using the proposed itemset representation. For any frequent item a_i, all the possible patterns a_i can be obtained by transversing the a_i's node-links, starting from a_i's head. Example 1 in Figure 4-12, performs the construction of the proposed graph-based algorithm. According to Apriori property: "if a pattern a is frequent, then all of its subsets are frequent". Therefore, if the pattern $(a_i{:}s_j)$ is compressed by the proposed representation then all the combinations of the items$\{a_1{:}s_1,\ a_2{:}s_2, a_3{:}s_3, ..., \ a_n{:}s_n\ \}$ are frequent. The redundant patterns are removed before adding them as a node in the graph. In Example 1, if the node {1} appears 5 times and its path includes {12:3,1:3}. In this path, {1,3,12} appears 2 in the database. At each level k, the frequent itemsets are joined with the same cardinality in level $k-1$ to produce the next candidate itemsets. The node {1} and {3} at level k_1, are joined to produce {1,3 :2} at level k_2 that appears 2 times in the dataset. Thus, if the itemset {1,3} is frequent then its subset i.e., {1} and {3} are also frequent. Similarly, for the frequent itemset {1, 3, 12, 31} all its subsets are also frequent. Thus, all the possible frequent itemsets are generated using the proposed graph representation.

4.7 Proof of Completeness

Lemma 1.2 Given a database D, all the possible sets of items of all the transactions can be extracted from the proposed graph-based algorithm
Proof. According to the construction of a graph-based algorithm, all the items in each transaction D are mapped to one path in the graph. Let the path be p_1, p_2, p_3, ..., p_n starting from the root leading to the node. c_{p_n} be the count of the node p_n and the cn_{p_n} be the count of all the children of the node p_n. According to the construction of the proposed graph algorithm, the path can be transverse from the root to the node p_n. Thus, the proposed graph-based algorithm maintains all the items projections from the transaction removing the duplicate nodes.
Hence, the proposed graph-based algorithm maintains the complete set of all items projected for each transaction of D without any duplication.

4.8 Discussion and Conclusion

The time consumption using the different rule mining algorithms for the different datasets is shown in Table 4.1. It is observed that the proposed algorithm surpasses the other algorithms in terms of time consumption. Furthermore, the memory used is also reduced since each item in the itemset takes only one bit of memory. The associations of subset and superset are maintained in the graph in such a way that the itemsets whose support count is less than the threshold value can easily be deleted. Thus, the proposed graph-based algorithm outperforms other algorithms. This shows that the algorithm is also beneficial, time-saving, and cost-efficient in different domains.