Performance Comparison of Association Rule Mining algorithms on Different Datasets using Tanagra Tool

Pushpa Devi
Department of Computer Science, Himachal Pradesh University, Summerhill, Shimla-5

Kishori Lal Bansal
Professor, Department of Computer Science, Himachal Pradesh University, Summerhill, Shimla-5

ABSTRACT

Today is the world of digital data. Every information or data is available online on internet or online resources. Data mining is a process or technique which helps to get useful information or related patterns out of huge amount of data. There are various techniques that are used for data mining. Association Rule Mining (ARM) is one of the most popular technique among all the data mining techniques. ARM technique is used to find out the correlation among the various itemsets present in the dataset. This study is focused on studying and comparing the performance of the association rule mining algorithms present in the Tanagra 1.4.50 data mining tool. Comparison of Apriori, Apriori PT, Assoc Outlier, Frequent Itemsets, Spv Assoc Rule and Spv Assoc Tree algorithms of association component in Tanagra Tool is done in this study. The performance is analyzed based on the execution time, memory usage and Number of rules generated for different number of instances, support and Rule Length. Three different datasets are used for comparison named as Chess Game, Mushroom and Hypothyroid. Results of three datasets show that Apriori algorithm take more memory than other algorithms. Association Outlier and Apriori algorithm take almost same amount of execution time. Frequent Itemsets, Apriori PT and Apriori algorithms generates more rules than other algorithms. The Spv Assoc Rule and Spv Assoc Tree are the special algorithms which are used to filter the rules based on the discrete class value and require less memory space, less execution time and generated few rules i.e. filtered rule than other algorithms present in the Tanagra tool.

Keywords: Apriori, Apriori PT, Assoc Outlier, Spv Assoc Rule, Spv Assoc Tree.

1. INTRODUCTION

Data mining is the process or technique of extracting useful knowledge, trends and patterns from large datasets by applying or posing various queries [1]. There are various techniques used for mining Data named as Classification, Clustering, Regression, Sequential Patterns, Prediction and Association Rules. Among all the techniques Association Rule is the most popular. Association Rules are simple **IF/THEN** statements where **if** part defines condition and **then** defines the results. Support and confidence are the two main measures of association rules. Support gives information about how frequently the items appear in the dataset and Confidence gives indication about the number of times **if-then** statements are found true [14]. Various algorithms are available for association rule mining, some of them are Apriori, Apriori PT, Frequent Itemsets etc. The main objective of paper is to compare the performance of Apriori, Apriori PT, Assoc Outlier, Frequent Itemsets, Spv Assoc Rule and Spv Assoc Tree algorithms. The mentioned algorithms are available in the Tanagra 1.4.50 Tool.

Brief Description about Association Rule Mining Algorithms:

- **Apriori:** Apriori algorithm is proposed in 1994 by R AGRAWAL AND R SRIKANT. As the name implies Apriori uses prior information of frequent itemsets properties [9]. Apriori algorithm is highly efficient in generating frequent item sets. To generate frequent item sets Apriori algorithm uses a "**bottom-up**" approach. Frequent subsets in apriori algorithm are extended Level-by-Level one item at a time and this step is known as **Candidate Generation** [2]. The algorithm will stop when there are no further successful extensions available.
- **Apriori PT:** The major drawback of Apriori algorithm was that it needs large memory space. A new approach is added into TANAGRA tool, the launching and the control of an external program in the form of Apriori PT algorithm [15]. at the time of the execution a temporary file is created and this temporary file is then transmitted to the "Christian BORGELT's" APRIORI.EXE exe file and rules will be automatically downloaded and displayed in tool.
- **Association Outlier:** Association outlier is used to build rules from an attribute value dataset. This component uses the association rule mining principle in order to detect outliers. At least two discrete attributes must be available in the dataset. This component can also handle binary 0/1 continuous attributes [6].
- **Frequent Itemsets**: The set of items that frequently appears in a transactional database or datasets is called as Frequent Itemsets. This is one of the main objectives of Data Mining. For example, if 80% of

the customers buys butter and bread, they will purchase Milk too. This Algorithm is based on Borgelt's "Apriori.exe" program [16].

- **Spv Assoc Rule:** is a supervised association rule generator. This algorithm computes all the rules leading to the discrete target attribute using apriori approach. The association rules generated by Spv Assoc Rule generation algorithm are limited to itemsets that consist of the Dependent Variable (DV). Tanagra tool has specificity to compare the predictable approaches, it can denote the class value "dependent variable = value" that desire to forecast [6].
- **Spv Assoc Tree** [18]**:** SPV ASSOC TREE or Supervised Association Tree allows characterizing a subset of examples with the conjunction of variables. The procedure uses internally a search tree but the outputs are rules. This is a supervised like association rule generation algorithm where the consequent of the rule is defined. CLASS VALUE enables you to define the subgroup which you want to characterize. The results are similar to the results of GROUP CHARACTERIZATION component, but this component give more detailed results. The main contribution of Spv Assoc Tree is that it can handle the interaction between two or more attributes.

2. LITERATURE REVIEW

M. Sharma, J. Choudhary and G. Sharma [3] evaluated the performance of Apriori and Predictive Apriori association rule mining algorithms based on the statistical of the datasets. The performance is compared using the **weka3.7.5** tool based on interesting measures, then for different datasets various statistical measures are calculated. After that based on the comparison of algorithms and statistical measures of datasets, using **see5** tool new rules are generated. Authors concluded that Predictive Apriori algorithm performs better than Apriori algorithm. **J. Nahar, T. Imam, K.S. Tickle, and Y.P.P. Chen** [4] did a study on association rule mining to detect factors which contribute to heart disease in males and females on UCI Cleveland dataset. The algorithms used for rule generation were Apriori, Predictive Apriori and Tertius. Resting ECG being either normal or hyper and slope being flat are potential high-risk factors for women only. For men, only a single rule expressing resting ECG being hyper was shown to be a significant factor. For women, resting ECG status is a key distinct factor for heart disease prediction. **J. Ashri** [5] did a study on the performance of the multidimensional database. Author has used the Apriori algorithm as the base algorithm and optimized the results using empirical algorithmic approach. WEKA, MATLAB, MySQL Database Engine and MySQL J Connector for JDBC tools are used for implementation. **S. Vijayarani and R. Prasannalakshmi** [7] compared the performance of Frequent Pattern Tree Growth algorithm and APRIORI Map/Reduce algorithms. Performance measures are execution time and number of association rules generated. Authors concluded that the performance of FP-Tree Growth algorithm is more efficient than APRIORI Map/Reduce algorithm. **S. Kumar, and K. Kaur** [8] reviewed different scope of data mining in future. Authors concluded that as data mining is the hub for an intelligent management decision support, a combination of high performance, scalable hardware and software is required for its successful implementation, which organizations can easily integrate with existing systems. So that users found it easy to implement and can use data mining to improve their decision-making. **D. Ai, H. Pan, X. Li, Y. Gao, and D. He** [10] presented a general survey of various association rule mining algorithms that are applicable to high-dimensional datasets. Authors has concluded that many methodologies are not able to manage the issue of high dimensionality. Many of the methods with their merits and faults have been proposed but new algorithms that can better balance scalability and interpretability are still not found. **K. Vijayalakshmi, S. Dheeraj and B.S.S. Deepthi** [11] build a prototype intelligent thyroid Prediction System by utilizing Big data and Information Mining strategies. Authors has used a Naïve Byes hypothesis to predict patients with Hypothyroid. Authors concluded that for intelligent thyroid system the naïve base classifier has given the best outcomes in provisions of accuracy and least execution time. **C. Wang, Y. Liu, Q. Zhang, H. Guo, X. Liang, Y. Chen, M. Xu, and Y. Wei** [12] proposed a novel parameter adaption strategy, which could incorporate promising Scale Factor(F) and Crossover Rate (Cr) pairs extracted by using Association Rule Mining into Differential Evolutionary (DE) algorithms. The experimental results of this study demonstrate that proposed Association Rule Mining – based parameter-based parameter adaptive strategy is able to enhance the performance of some Differential Evolutionary variants. **P. Devi, V.S. Bhardwaj and K.L. Bansal** [13] has compared three ARM algorithms Apriori, Apriori PT and Frequent Itemsets using the Tanagra Tool. The parameters used are Execution Time (in ms) and Memory usage (in kb) for varying support value, no. of instances and rule length. Authors concluded that when support value is increased the Apriori algorithm takes less execution time than other two algorithms. When the no. of instances reduced to one third of the total instances Frequent Itemsets outperforms well both in case of memory and execution time. When rule length is increased the Apriori algorithm performs better than other two algorithms.

3. METHODOLOGY USED

This research is based on the ongoing research in data mining field. The Research Methodology followed for this study basically includes two steps:

1. **Theoretical Study:** The theoretical study includes the study of Data Mining concepts and techniques, association rule mining algorithms and overview of Tanagra Tool. Major contributors of this study are Research papers and Internet.
2. **Experimentation:** Tanagra 1.4.50 is the open source tool used for the experimental purpose in which A-priori, A-priori PT, Association Outlier, Frequent Itemsets, Spv Assoc Rule and Spv Assoc Tree association rule mining algorithms compared. Three different datasets are used named as: **Chess** (https://www.openml.org/d/3 (downloaded on 02-03-2020 1:30 PM)) having 3196 instances and 37 attributes, **Hypothyroid** (https://www.openml.org/d/41946 (downloaded on 25-04-2020 PM 4:30 PM)) having 3772 instances and 30 attributes and Mushroom (https://www.openml.org/d/24 (downloaded on 02-03-2020 4:40 PM)) having 8124 instances and 23 attributes.

4. ANALYSIS OF RESULTS

The results are analyzed based on three parameters execution time, memory used and number of rules generated. Execution time (in micro-seconds(ms)) of the algorithms and number of rules generated by an algorithm is provided by the tool itself and operating system information is used for measuring the memory usage (in mega-bits(mb)) [34]. The following tables present the test results of the A-priori, A-priori PT, Assoc Outlier, Frequent Itemsets, Supervised Association Rule (Spv Assoc Rule) and Supervised Association Tree (Spv Assoc Tree) for different datasets named as Chess, Hypothyroid and Mushroom. The tables show the test results and figures shows test results of the tables in graphical form.

Dataset 1: Chess Game

1. **For varying support**

Table 1.1 Runtime in ms

Support	Runtime in ms					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	1422	32	1422	47	63	109
0.43	844	31	906	47	0	31
0.53	516	47	516	78	0	0

From Table 1.1 it can be stated that **Apriori and Assoc Outlier are two algorithms in which the execution time is decreasing when support value is increasing**. The results of these two algorithms are almost same.

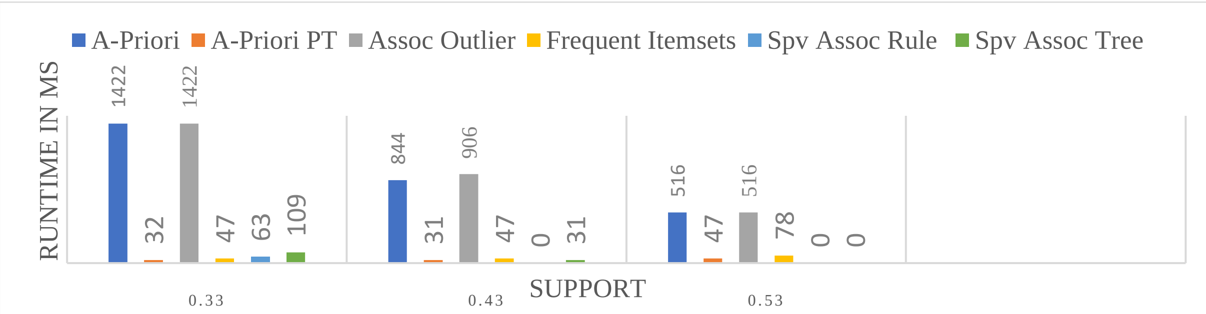

Fig 1.1 Support vs. Runtime

Table 1.2 Memory usage in mb

Support	Memory usage in mb					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	163.4	32.4	42.9	14.4	10.1	9.5
0.43	96.6	24.8	32.3	11.2	9.3	9.6
0.53	50.0	17.2	19.5	9.9	9.3	9.5

The test result of Table 1.2 shows that the amount of memory that above-mentioned algorithms occupy decreased with the increase in the support value. As a result of this table it can be concluded that **for varying support value different algorithms take less memory than others**.

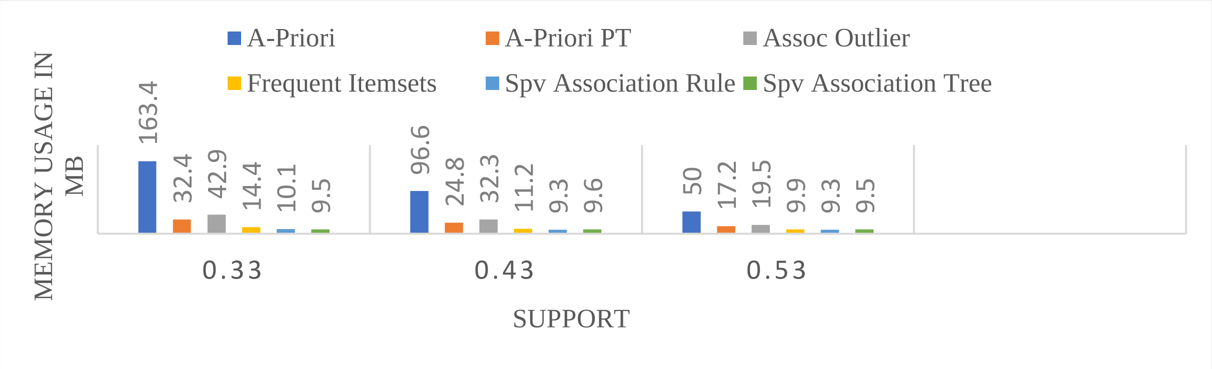

Fig 1.2 Support vs. Memory usage

Table 1.3 Number of rules generated

Support	**Number of rules generated**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	29018	166363	149	69289	18	8
0.43	16540	112762	80	37038	0	0
0.53	7843	63287	34	19011	0	0

From the table 1.3 it can be stated that Apriori PT and Frequent Itemsets are the two algorithms which are generating the huge number of rules. **The number of rules generated by all the algorithms decreased with the increase in support value.**

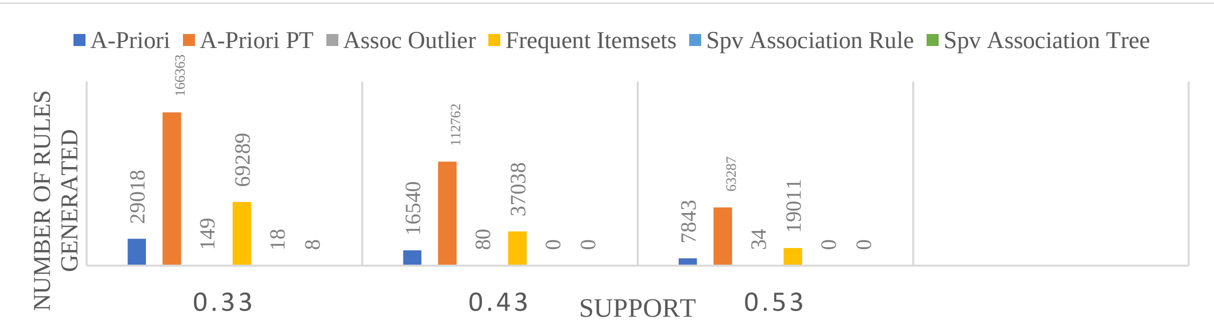

Fig 1.3 Support vs. Number of rules generated

2. **Number of instances**

Table 2.1 Runtime in ms

Number of instances	Runtime in ms					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3196	1422	32	1422	47	63	109
1598	1219	32	1141	31	188	110
799	1125	32	1109	16	188	110

Table 2.1 shows that execution time for Apriori, Assoc Outlier and frequent itemsets algorithms reduced with the reduction in the number of instances and increased in case of Supervised Association Rule and Supervised Association Tree Algorithms. In case of Apriori PT algorithm Execution time remain constant with the reduction in number of instances.

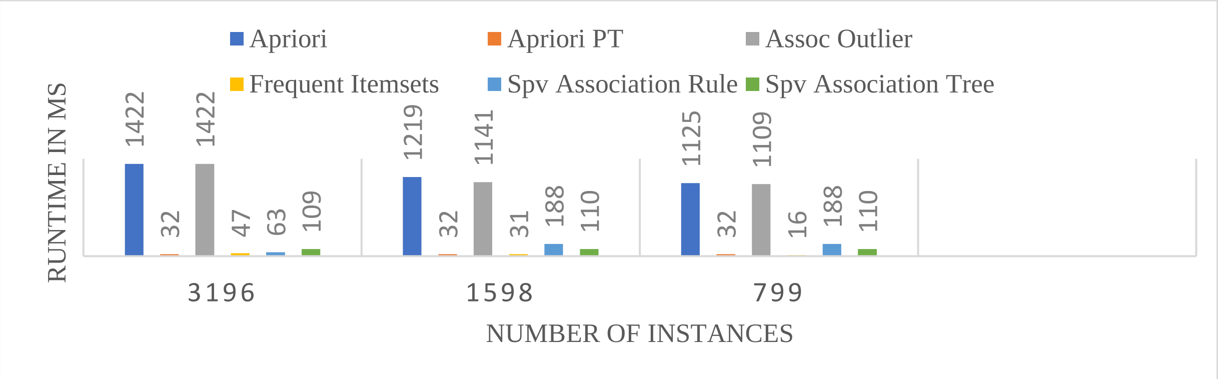

Fig 2.1 Number of instances vs. Runtime

Table 2.2 Memory usage in mb

Number of instances	Memory usage in mb					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3196	163.4	32.4	42.9	14.4	10.1	9.5
1598	136.8	39.3	88.6	14.9	9.8	9.8
799	138.6	40.1	88.0	14.6	9.8	9.8

The test result of Table 2.2 shows that there is not much difference in the amount of memory above mentioned algorithms require with the reduced number of instances. **Apriori algorithm the one algorithm which takes more memory than other algorithms.** As a result of this table it can be concluded that when number of instances decreased different algorithms take less memory than others.

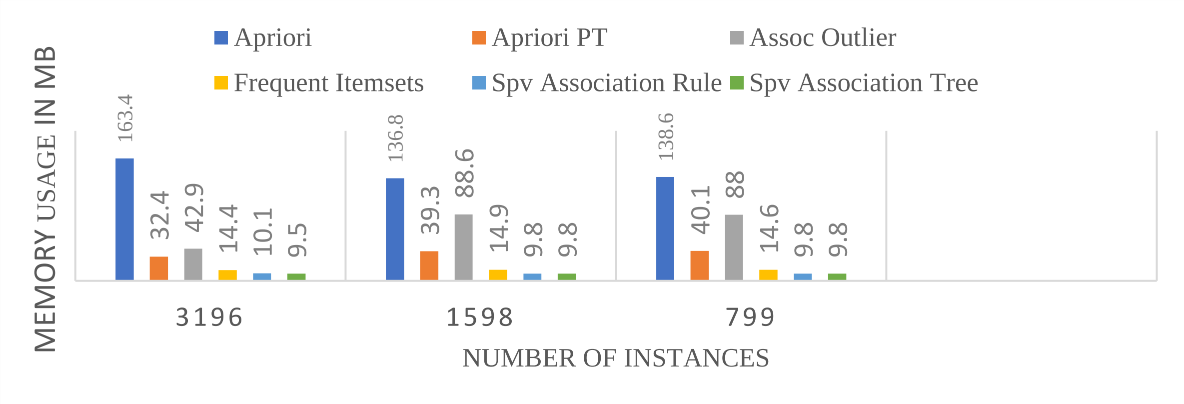

Fig 2.2 Number of instances vs. Memory usage

Table 2.3 Number of rules generated

Number of instances	**Number of rules generated**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3196	29018	166363	149	69289	18	8
1598	23766	210443	1017	72594	74	75
799	23766	210443	1017	72594	74	75

From the table Table 2.1 it can be concluded that Apriori PT algorithm has generated the highest number of rules with the reduction in number of instances.

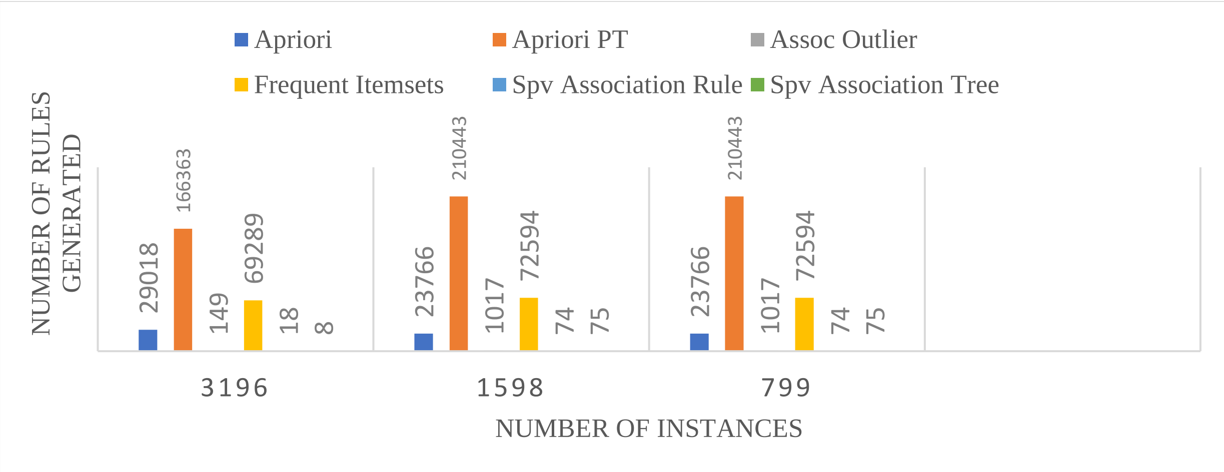

Fig 2.3 No. of instances vs. Number of rules generated

3. Rule Length

Table 3.1 Runtime in ms

Rule Length	**Runtime in ms**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree

3	93	31	94	47	0	31
4	1422	32	1422	47	63	109
5	30047	32	30875	32	500	578

Table 3.1 shows that as the rule length is increased the runtime or the execution time also increased. Frequent itemset is the only algorithm whose execution time has not increased with the increase in the rule length. So, from the table above it can be **concluded that in most of the cases execution time increase with the increase in the rule length.**

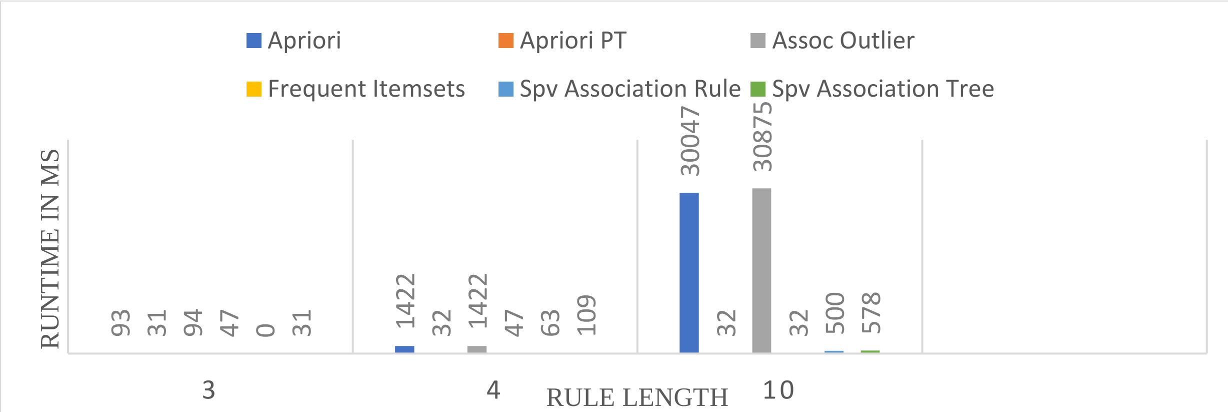

Fig 3.1 Rule length vs. Runtime

Table 3.2 Memory usage in mb

Rule Length	**Memory usage in mb**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	15.6	14.2	10.4	11.9	11.8	10.5
4	163.4	32.4	42.9	14.4	10.1	9.5
5	482.0	176.9	278.5	44.03	11.8	11.6

The test result of Table 3.2 shows that when rule length is increased, memory usage also increased in case of Apriori, Apriori PT, Assoc Outlier and Frequent Itemsets algorithms. In case of Spv Assoc Rule and Spv Assoc Tree Memory usage almost remain same. So, from the above table it can be **concluded that when rule length is increased the memory usage also increase and apriori take the highest memory among all algorithms.**

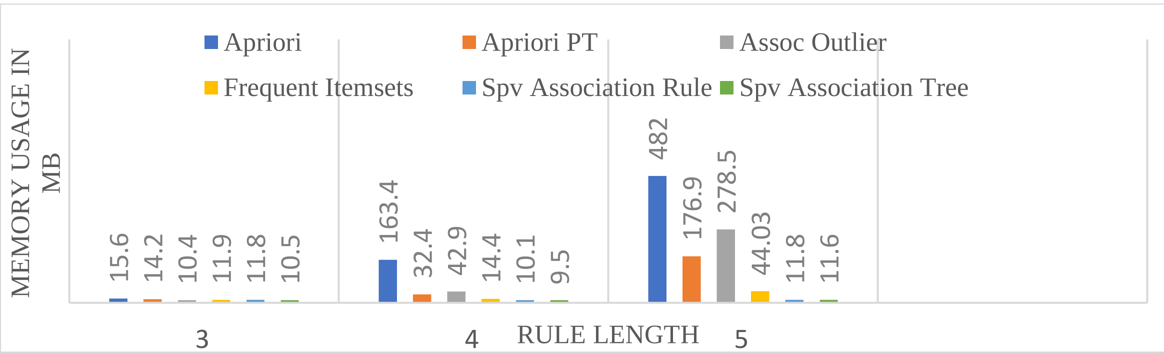

Fig 3.2 Rule length vs. Memory usage

Table 3.3 Number of rules generated

Rule Length	Number of rules generated					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	1516	17180	6	8797	1	1
4	29018	166363	149	69289	18	8
5	342923	1035327	1691	260879	137	137

Table 3.3 shows that the number of rules generated by all algorithms increase with the increase in the rule length. So, it can be concluded that **if rule length is increased the number of rules generated by the algorithms also increase and Apriori PT generates the highest number of rules**.

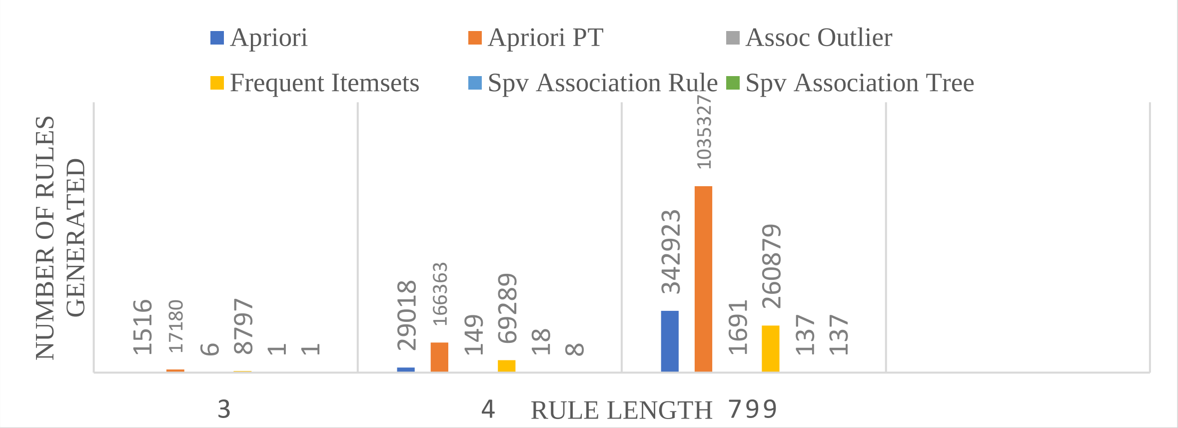

Fig 3.3 Rule length vs. Number of rules generated

Dataset 2: Hypothyroid

1. **For varying support**

Table 1.1 Runtime in ms

Support	Runtime in ms					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	359	32	409	47	0	0
0.43	359	63	390	47	0	0
0.53	375	47	344	47	0	0

From the table 1.1 it can be concluded that there is a variation in the time required by each algorithm for execution, some algorithms require same amount of time and other algorithms have a variation in execution time with increased support value. **The one algorithm which took the highest execution time is Apriori algorith**

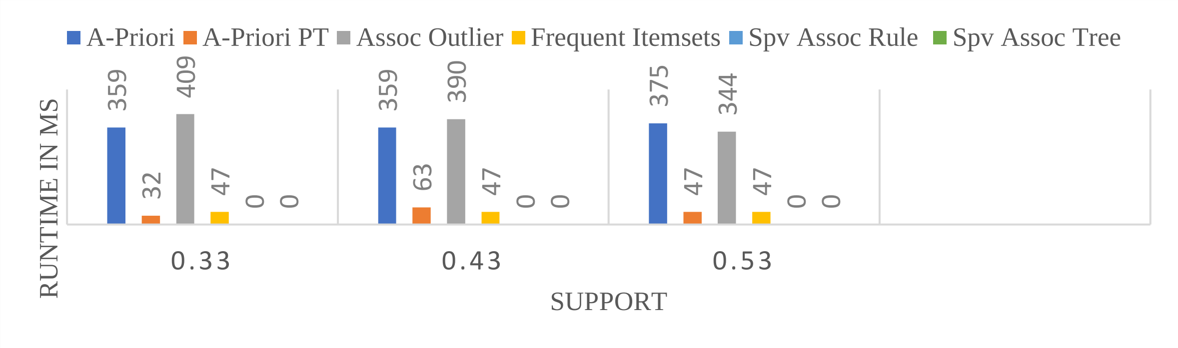

Fig 1.1 Support vs. Runtime

Table 1.2 Memory usage in mb

Support	**Memory usage in mb**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv AssocTree
0.33	30.0	18.1	28.9	13.9	13.5	13.1
0.43	29.6	18.0	28.5	13.6	13.1	13.1
0.53	28.2	15.1	25.8	13.6	13.1	13.1

Table 1.2 shows that there is not much difference in the amount of memory occupied by the above-mentioned algorithms. All the above algorithms almost require the same amount of memory with the increase of the support value. From the table above it can be concluded that Apriori algorithm is the one algorithm is one algorithm which took the highest amount of memory.

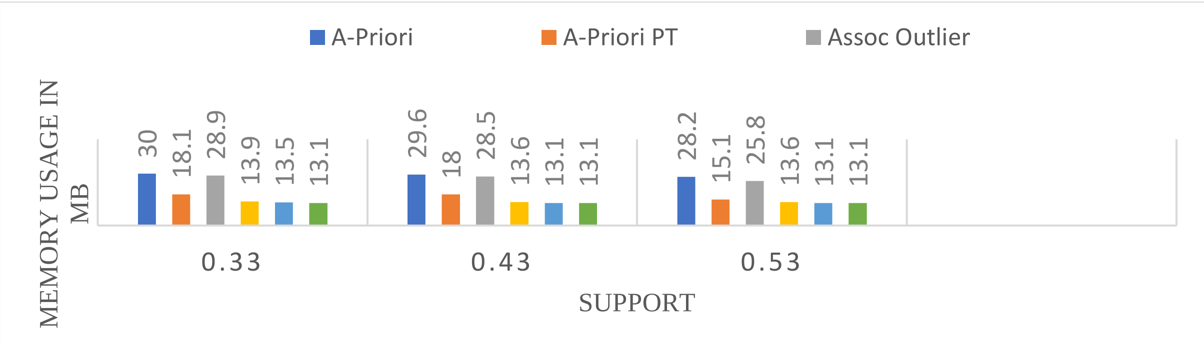

Fig 1.2 Support vs. Runtime

Table 1.3 Number of rules generated

Support	**Number of rules generated**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	1829	44715	400	12871	0	0
0.43	1823	43637	390	12359	0	0
0.53	1748	39845	369	11049	0	0

From the table 1.3 it can be stated that Apriori PT is one algorithm which generated huge number of rules. **The number of rules generated by all the algorithms other than Spv Association and Spv Association Tree (generated 0 rules) decreased with the increase in support value.**

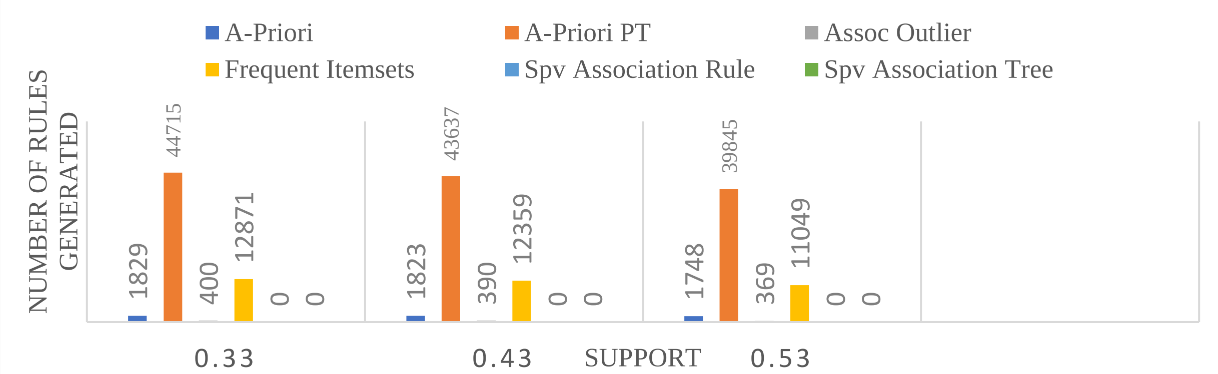

Fig 1.3 Support vs. Number of rules generated

2. Number of instances

Table 2.1 Runtime in ms

Number of instances	Runtime in ms					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3772	359	32	409	47	0	0
1886	219	31	250	47	0	0
943	172	15	156	15	0	0

Table 2.1 shows that the execution time is reduced with the decrease in the number of instances. In case of Spv Assoc Rule and Spv Assoc Tree execution time is zero as there is no rule generated. **From the table above it can be concluded that when the number of instances decreased the execution time also decrease**.

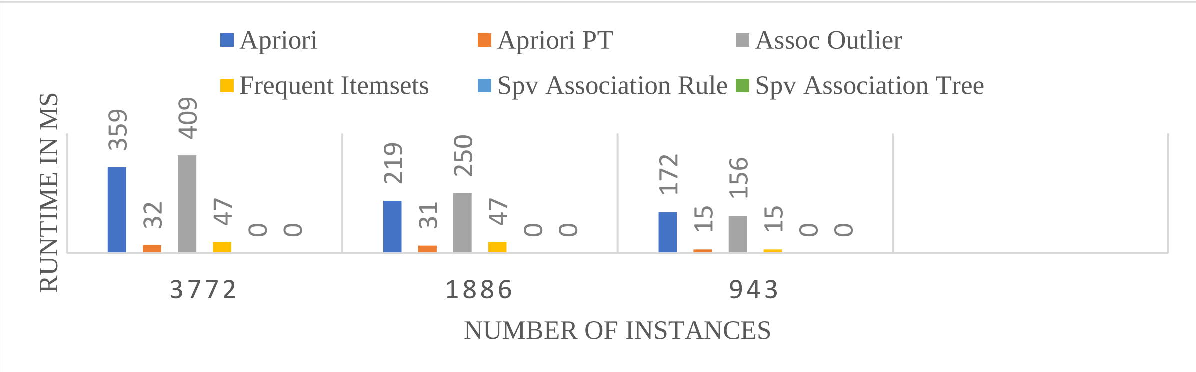

Fig 2.1 Number of instances vs. Runtime

Table 2.2 Memory usage in mb

Number of instances	Memory usage in mb					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3772	30.0	18.1	28.9	13.9	13.5	13.1

1886	25.1	15.2	26.4	14.4	14.4	14.3
943	25.7	14.6	27.1	14.4	14.3	14.2

Table 2.2 shows that the memory required by all above mentioned algorithms is almost same when the number of instances are decreased. From the above table it can be **concluded that Apriori algorithm and Association Outlier algorithm took the more time than other algorithms**.

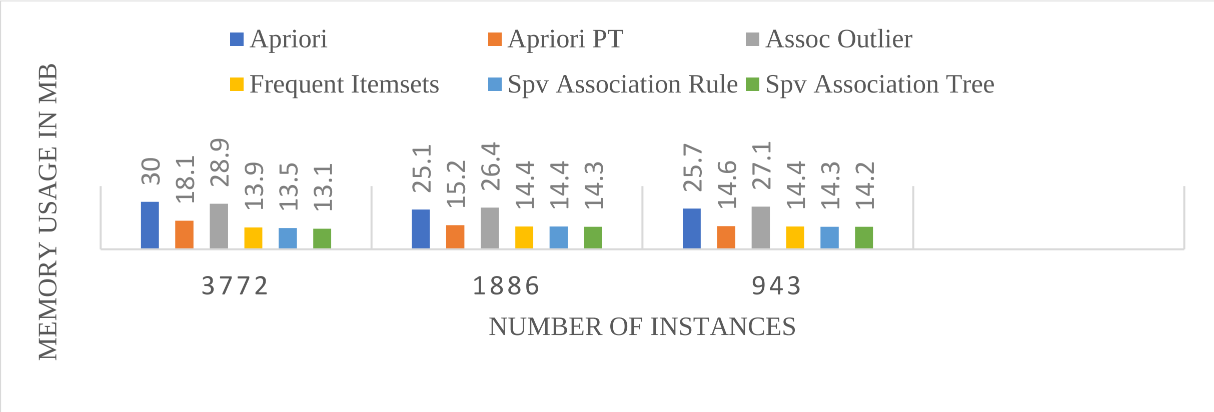

Fig 2.2 Number of instances vs. Memory usage

Table 2.3 Number of rules generated

Number of instances	**Number of rules generated**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3772	1829	44715	400	12871	0	0
1886	1673	37314	361	10821	0	0
943	1342	37276	295	10789	0	0

Table 2.3 shows that the number of rules generated by the algorithms decreased with the decrease in the number of instances and there are no rules generated by the Spv Assoc Rule and Spv Assoc Tree algorithms. From the above table it can be **concluded that the number of rules generated by above algorithms decrease with decrease in the number of instances.**

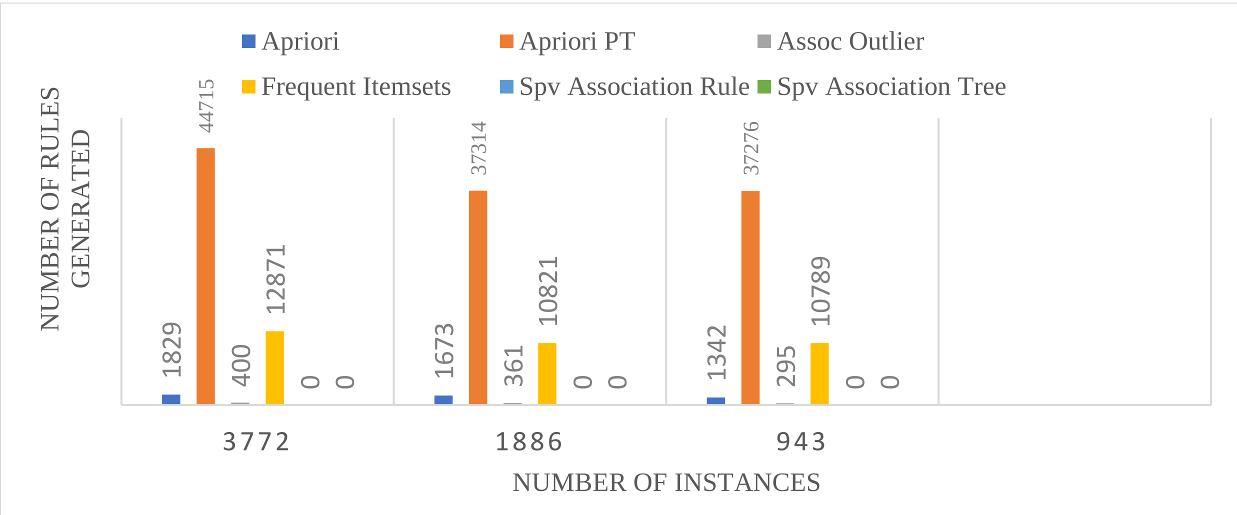

Fig 2.3 Number of instances vs. Number of rules generated

3. Rule Length

Table 3.1 Runtime in ms

Rule Length	Runtime in ms					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	62	47	31	47	0	0
4	359	32	409	47	0	0
5	3812	63	3859	31	16	16

From the table 3.1 it can be stated that the execution time is increasing with the increase in the rule length. So, from the table above it can be concluded that **Apriori and Assoc outlier are the two algorithms which took the highest runtime than other algorithms and execution time increased with increase in rule length.**

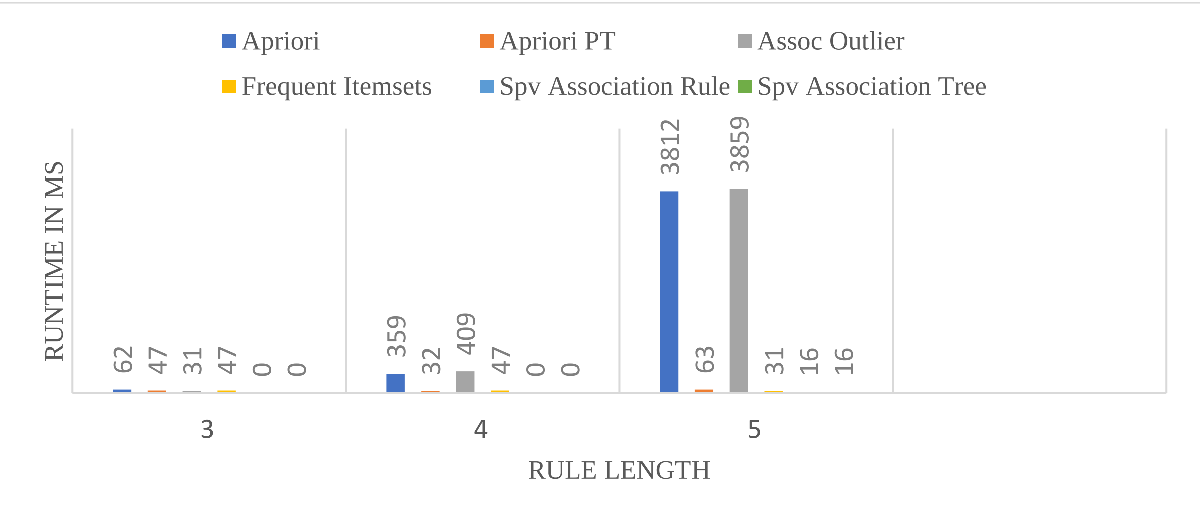

Fig 3.1 Rule Length vs. Runtime

Table 3.2 Memory usage in mb

Rule Length	Memory usage in mb					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	14.4	14.5	14.4	14.3	14.2	14.2
4	30.0	18.1	28.9	13.9	13.5	13.1
5	182.6	63.5	193.0	19.6	14.2	14.2

Table 3.2 shows that the memory usage is increasing as the length of rule is increased. So, from the table it can be stated **that Apriori and Assoc Outlier require more memory than the other algorithms.**

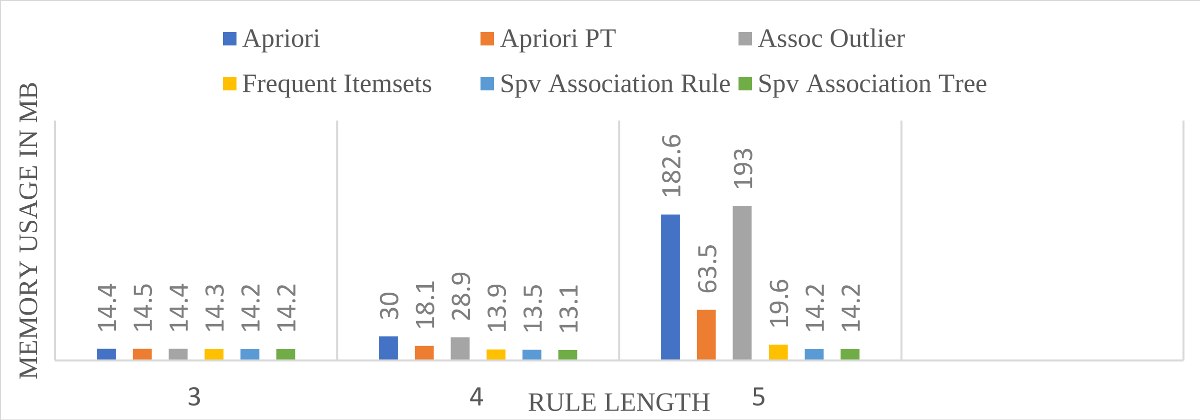

Fig 3.2 Rule Length vs. Memory usage

Table 3.3 Number of rules generated

Rule Length	Number of rules generated					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	88	6050	29	2299	0	0
4	1829	44715	400	12871	0	0
5	23962	636217	3482	54654	0	0

Table 3.3 shows that the number of rules generated by the above algorithms increased as the rule length increased and there are no rules generated by the Spv Assoc Rule and Tree algorithm. So, it can be **concluded that as the rule length increase number of rules generated by the algorithms also increase and Apriori PT generated the highest number of rules.**

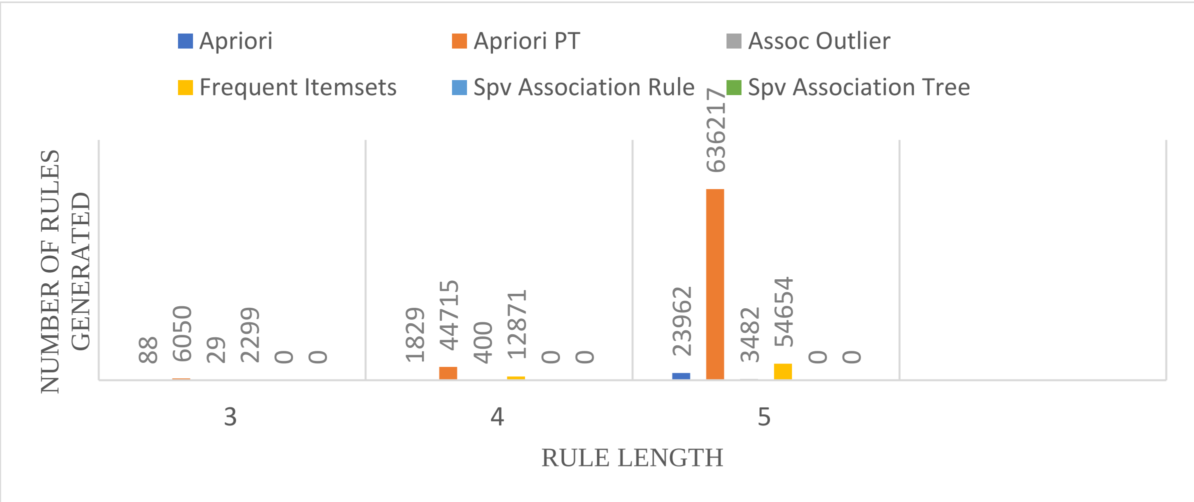

Fig 3.3 Rule Length vs. Number of rules generated

Dataset 3: Mushroom

1. **For varying support**

Table 1.1 Runtime in ms

Support	**Runtime in ms**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	47	78	47	78	46	16
0.43	31	62	15	62	0	16
0.53	0	62	0	62	0	0

From the table1.1 it can be **concluded that execution time is decreasing with the increase in support value and Apriori PT and Frequent Itemsets algorithms took the same amount of execution time for all the three support values**.

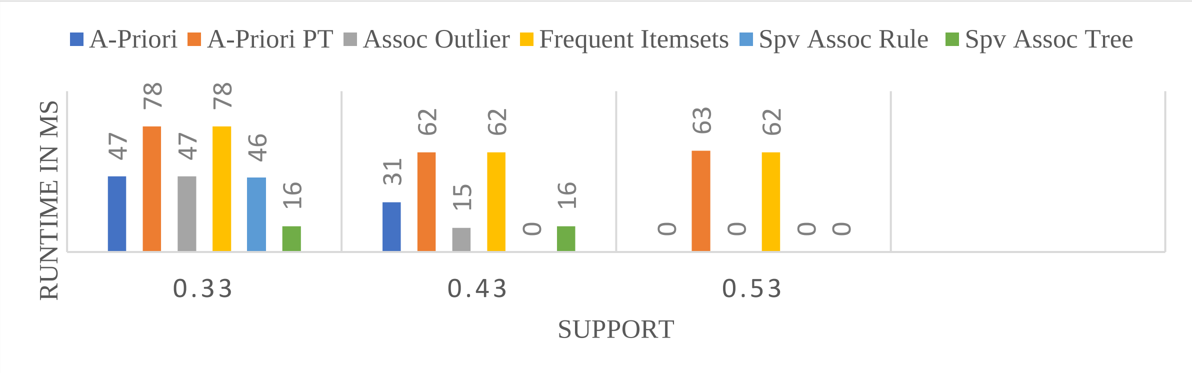

Fig 1.1 Support vs. Runtime

Table 1.2 Memory usage in mb

Support	**Memory usage in mb**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	22.3	19.7	11.3	11.6	11.5	11.6
0.43	20.0	19.8	11.2	11.8	11.6	11.6
0.53	19.6	19.8	11.2	11.4	11.6	11.6

Table 1.2 shows the memory requirement of different algorithms when executed with varying support value. From the table above it can be concluded that **almost same amount of memory is required by all the algorithms and the two algorithms which require more memory than others is Apriori and Apriori PT.**

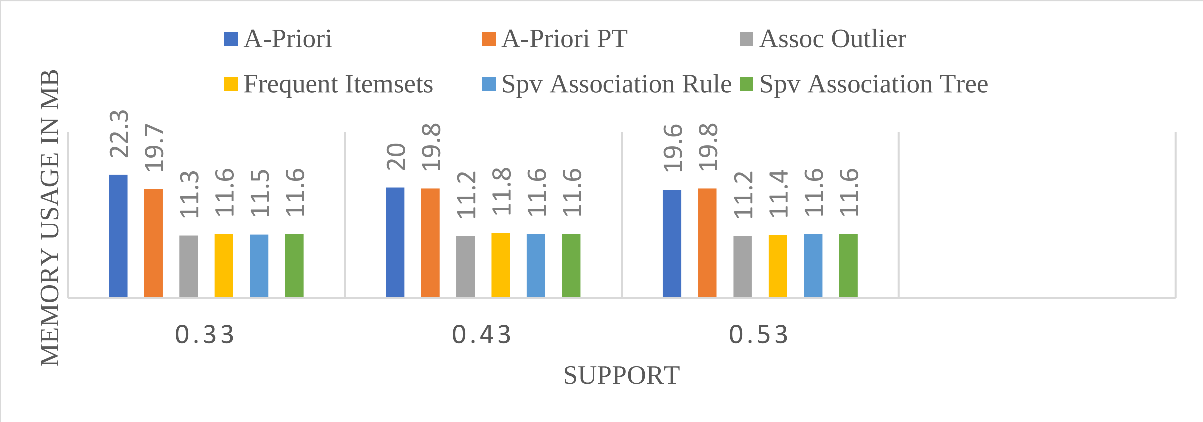

Fig 1.2 Support vs. Memory usage

Table 1.3 Number Of rules

Support	**Number of rules generated**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
0.33	1861	2375	0	1020	7	7
0.43	222	774	0	343	0	0
0.53	9	227	0	108	0	0

From the Table 1.3 above it can be **concluded that the number of rules generated are decreasing when support value is increased.**

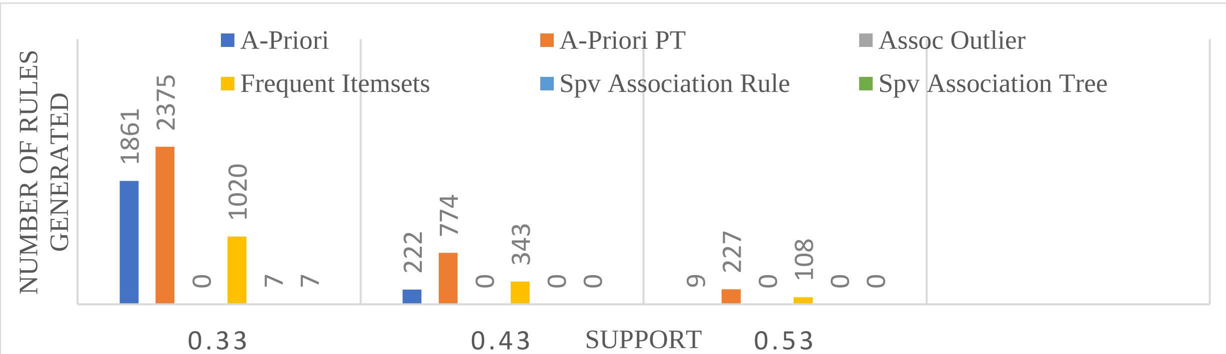

Fig 1.3 Support vs. Number of rules generated

2. Number of Instances

Table 2.1 Runtime in ms

Number of instances	**Runtime in ms**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
8124	47	78	47	78	46	16
4062	78	31	109	31	16	0

2031	47	31	62	16	0	0

From Table 2.1 it can be stated that when number of instances are reduced from 8124 to 4062 execution time of apriori and Assoc Outlier Algorithms has increased and when instances reduced from 4062 to 2031 the execution is decreased. The **execution time for other algorithms has decreased with the decreased in the number of instances.**

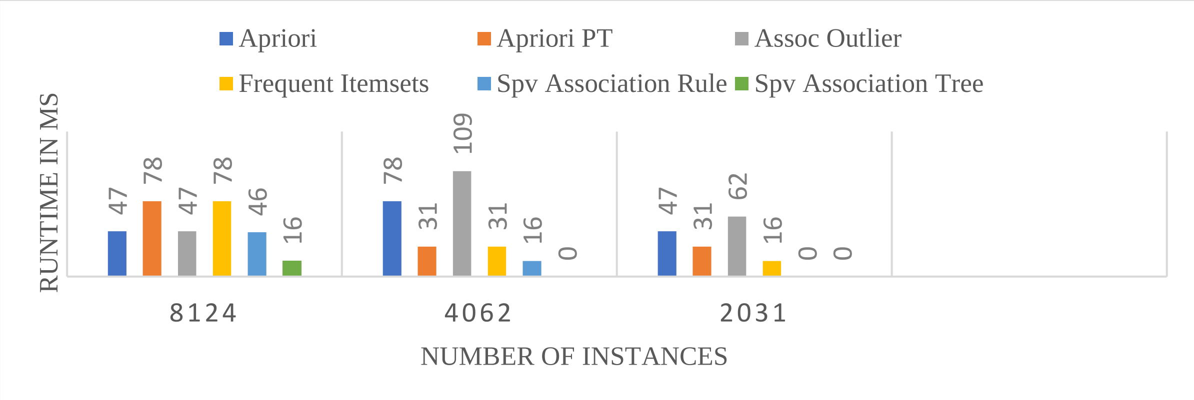

Fig 2.1 Number of instances vs. Runtime

Table 2.2 Memory usage in mb

Number of instances	**Memory usage in mb**					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
8124	22.3	19.7	11.3	11.6	11.5	11.6
4062	59.0	12.0	16.2	12.0	11.9	12.0
2031	58.7	12.2	12.4	12.4	12.3	12.3

Table 2.2 shows that there is not much difference in the amount of memory occupied by the different algorithms for different number of instances but in all the algorithms **Apriori is one algorithm which occupy more memory than other algorithms.**

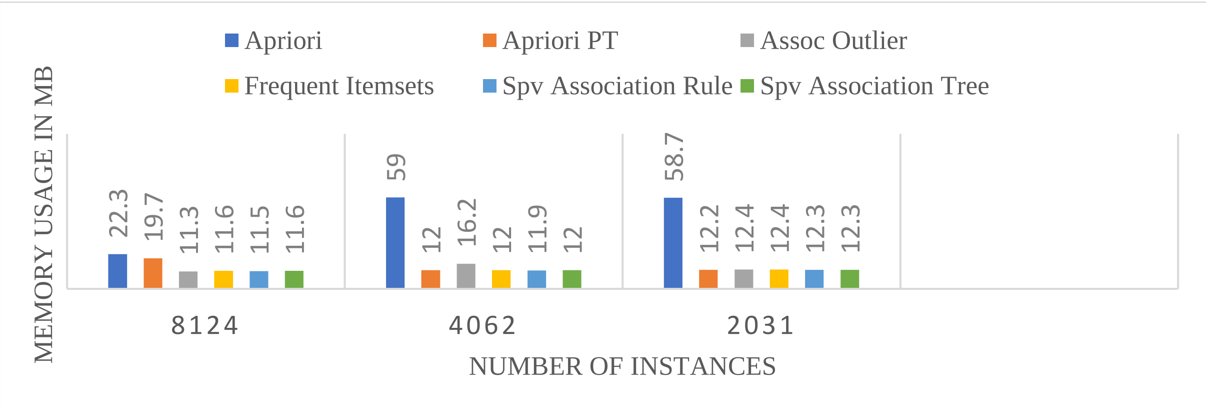

Fig 2.2 Number of instances vs. Memory usage

Table 2.3 Number of rules generated

Number of instances	Number of rules generated					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
8124	1861	2375	0	1020	7	7
4062	9593	8429	1375	3052	0	0
2031	9512	8537	612	3004	0	0

From Table 2.3 it can be stated that for different algorithms, number of rules being generated are increasing and decreasing with the decrease in number of instances.

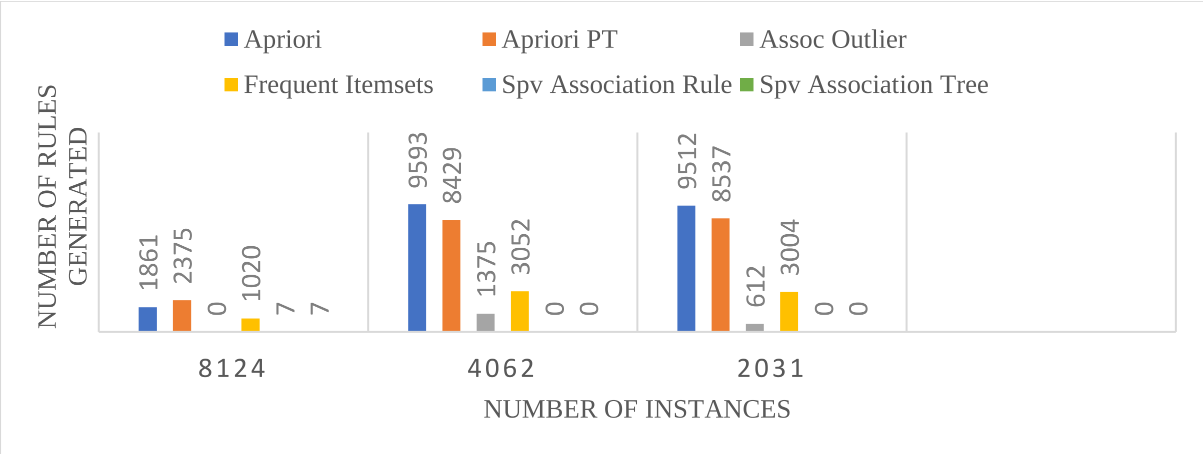

Fig 2.3 Number of instances vs. Number of rules generated

3. Rule Length

Table 3.1 Runtime in ms

Rule Length	Runtime in ms					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	31	62	31	78	15	0
4	47	78	47	47	46	16
5	79	47	47	63	79	15

From Table 3.1 it can be **concluded that the execution time or the runtime of different algorithms is increasing and decreasing with the increase in the length of the rule.**

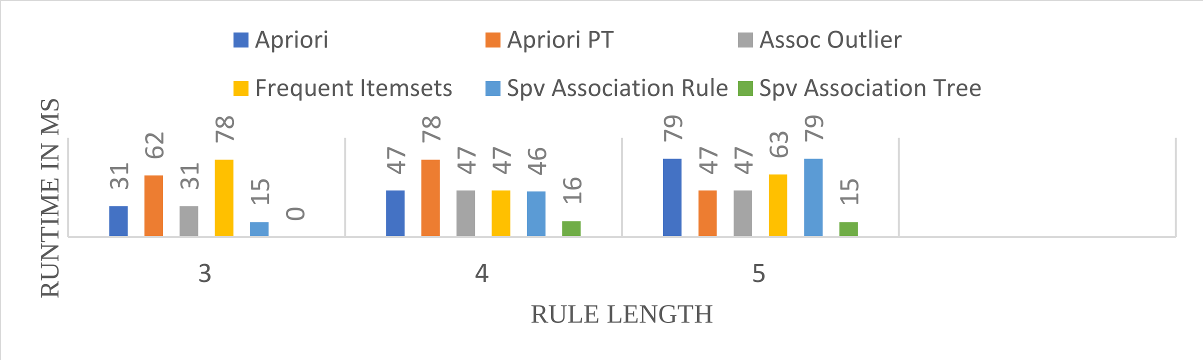

Fig 3.1 Rule Length vs. Runtime

Table 3.2 Memory usage in mb

Rule Length	Memory usage in mb					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	15.3	14.3	14.3	14.1	14.0	13.9
4	22.3	19.7	11.3	14.3	11.5	11.6
5	25.2	14.6	14.4	14.3	14.2	13.9

Table 1.3 show that with the increase in the rule length the memory required by different algorithms is also increasing and decreasing but in most of the cases requirement is increasing. From the table above it can be **concluded that the Apriori algorithm occupied more memory than other algorithms when rule length is increased.**

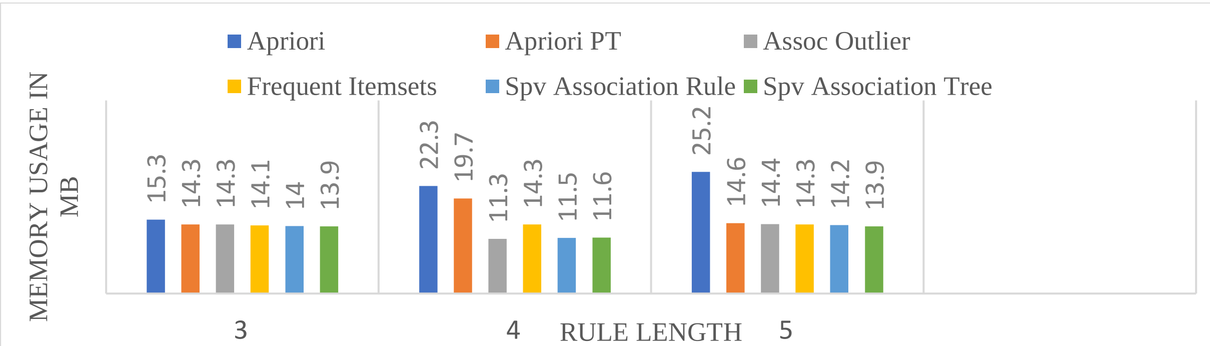

Fig 3.2 Rule Length vs. Memory usage

Table 3.3 Number of rules generated

Rule Length	Number of rules generated					
	Apriori	Apriori PT	Assoc Outlier	Frequent Itemsets	Spv Assoc Rule	Spv Assoc Tree
3	377	853	0	1020	1	0
4	1861	2375	0	421	7	7
5	3558	3044	0	1105	16	16

From the Table 3.3 it can be **concluded that number of rules generated by different algorithms has increased with the increase in rule length**.

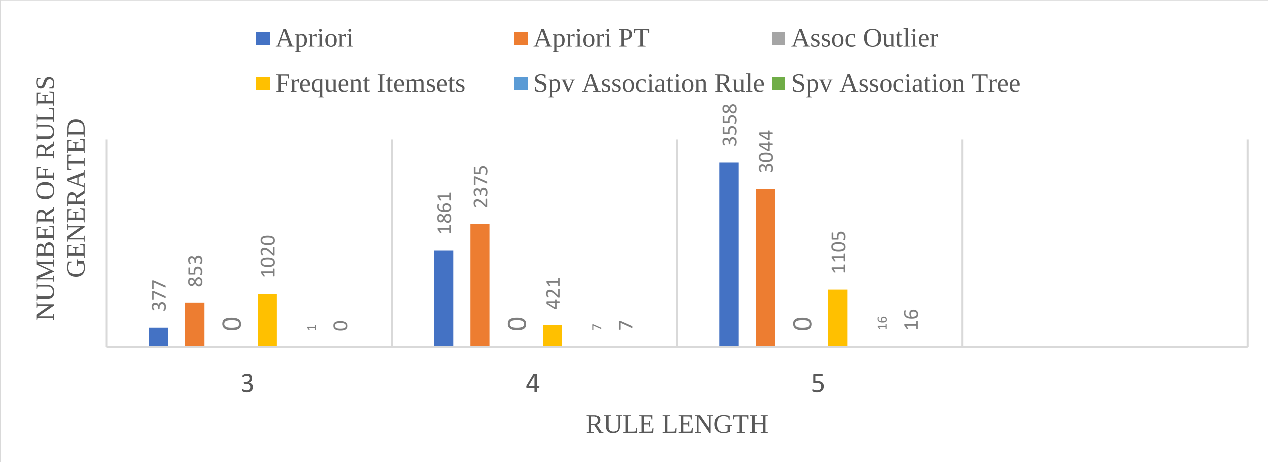

Fig 3.3 Rule Length vs. Number of rules generated

5. CONCLUSION

In this study six association rule mining algorithms namely, **A-priori**, **A-priori PT**, **Assoc Outlier**, **Frequent Itemsets, Spv Assoc Rule** and **Spv Assoc Tree** are compared. Parameters used are **execution time, memory used** and **Number of rules generated** for different number of instances, support and Rule Length. The comparison is done on three different datasets named as **Chess game, Mushroom** and **Hypothyroid**. When same parameter values are supplied to three different datasets from three different fields having different number of attributes and instances, the results of three datasets show that Apriori algorithm take more memory than other algorithms. Association Outlier and Apriori algorithm take almost same amount of execution time. Frequent Itemsets, Apriori PT and Apriori algoithms generates more rules than other algorithms. The Spv Assoc Rule and Spv Assoc Tree are the special algorithms which are used to filter the rules based on the discrete class value and require less memory space, less execution time and generated few rules i.e. filtered rule than other algorithms present in the Tanagra tool.

6. FUTURE SCOPE

Future work may include the development of new algorithm by combining the good points of all the algorithms used in this study into one algorithm and then comparing this new algorithm with the other algorithms presented in the TANAGRA TOOL with more performance evaluation parameters.

REFERENCES

[1] B. Thuraisingham, Data mining: technologies, techniques, tools, and trends. CRC press, 2014.

[2] K. Vanitha, "EVALUATING THE PERFORMANCE OF ASSOCIATION RULE MINING ALGORITHMS", Journal of Global Research in Computer Science, vol. 2, pp.101-103, June 2011.

[3] M. Sharma, J. Choudhary and G. Sharma, 2012, “Evaluating the Performance of Apriori And Predictive Apriori Algorithm to Find New Association Rules Based on the Statistical Measures of Datasets”, *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, Vol. 01, Issue 06, August 2012.

[4] J. Nahar, T. Imam, K.S. Tickle, and Y.P.P. Chen, “Association rule mining to detect factors which contribute to heart disease in males and females”, *Expert Systems with Applications*, Vol. 40, Issue 4, pp.1086-1093,2013.

[5] J. Ashri, “PERFORMANCE EVALUATION OF ASSOCIATION RULE MINING ON PARTITIONED DATA*”, International Journal of Enterprise Computing and Business Systems*”, Vol. 4, Issue 2, 2014.

[6] S. Vijayarani and R. Prasannalakshmi, “COMPARATIVE ANALYSISOF ASSOCIATION RULE GENERATION ALGORITHMSIN DATA STREAMS”, *International Journal on Cybernetics & Informatics (IJCI)*, Vol. 4, Issue 1, February 2015.

[7] S. Vijayarani and R. Prasannalakshmi, “Association Rule Generation in Data Streams using FP-Growth and APRIORI MR Algorithms”, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 3, Issue 9, pp.15-25, September 2015.

[8] S. Kumar, and K. Kaur, “Review of Data mining (Knowledge discovery) in the Future”, *International Journal of Advanced Research in Computer Science*, Vol.*7, Issue* 6,2016.

[9] M. Sinthuja, N. Puviarasan, and P. Aruna, “Evaluating the performance of association rule mining algorithms”, World Applied Sciences Journal, Vol. 35, Issue 1, pp. 43-53, 2017.

[10] D. Ai, H. Pan, X. Li, Y. Gao, and D. He, “Association rule mining algorithms on high-dimensional datasets”, *Artificial Life and Robotics*, Vol *23*, Issue 3, pp.420-427, 30 May 2018.

[11] K. Vijayalakshmi, S. Dheeraj and B.S.S. Deepthi, “Intelligent Thyroid prediction system using Big data”, *International Journal of Computer Sciences and Engineering*, Vol. 6, Issue 1, 31 Jan 2018.

[12] C. Wang, Y. Liu, Q. Zhang, H. Guo, X. Liang, Y. Chen, M. Xu, and Y. Wei, “Association rule mining-based parameter adaptive strategy for differential evolution algorithms”, *Expert Systems with Applications 123*, pp.54-69, 11 January 2019.

[13] P. Devi, V.S. Bhardwaj and K.L. Bansal, “Association Rule Mining on Spambase Dataset using Tanagra”, *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* Volume-9, Issue-6, April 2020.

[14] https://searchbusinessanalytics.techtarget.com/definition/associationrules-in-data-mining [23/10/2019 5:20 PM]

[15] http://eric.univlyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_A_Priori_ Prefix_Tree.pdf [23/10/2019 5:50 PM]

[16] http://eric.univlyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_Itemset_ Mining.pdf [23/10/2019 8:13 PM]

[17] http://eric.univlyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_Perfs_Co mp_SVM.pdf [23/10/2019 10:16 PM]

[18]http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_Spv_Assoc_Tree.pdf Accessed on : 12/04/2019 3:00 PM