International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Distributed FP Growth Algorithm for Cloud Platform without Exposing the Individual Transaction Data

Saritha Byreddi, A. Rama Mohan Reddy

***Abstract:** Data mining is a concept of extracting the required patterns to take appropriate decisions. One of the major challenges in data mining is to extract hidden patterns with the secure and privacy from the huge databases. Privacy preserving is a method used to extract hidden patterns with privacy. In this paper Mining Association rules with privacy preserving mechanism in the cloud platform is proposed. It is a powerful technique to find the hidden pattern in the distributed database. For now many mechanisms has proposed but it has many drawback, not proven and not specific. In cloud the data is stored in the servers. The data is distributed in different servers in cloud platform. Each server has one of the transaction data. The current paper proposed the distributed FP growth algorithm for cloud platform without exposing the individual transaction data. The results proved that the proposed algorithm is best to extract hidden pattern from Cloud platform in terms of efficiency.*

***Keywords:** Data mining, association rules, privacy preserving, patterns.*

I. INTRODUCTION

In the Data Mining the frequent pattern is very essential for discovery of knowledge. First this was done by the candidate generation algorithm called Apriori but the drawbacks in the Apriori is need a more memory for storing a candidate set and require multiple times of scanning the data base. By using the tree structured FP-Growth mining association algorithm. The drawbacks of Apriori can be solved currently the FP-based distributed mining algorithms are PFP-tree [10], FP-Forest [11], LFP-tree [12] and MLPT [13].

Association rule mining is one of the useful and popular method for finding the item sets and inherent regularities. This can be divided in to two phases those are finding the frequent item sets and another one is generating association rules from the frequent item set. FIM is the one of the most time consuming in association rule mining [6].

Manuscript published on 30 September 2019
* Correspondence Author

Ms Saritha Byreddi*, Research Scholar, Department of Computer Science and Engineering, Sri Venkateswara Univrsity, Tirupati.

Dr.A.Rama Mohan Reddy, Professor, Computer Science and Engineering, SV University College of Engineering, Tirupati,

Mining item sets plays major role in sequential pattern, emerging pattern, mining association, causality, correlation, episodes, partial periodicity, multimedia patterns and so many other data mining tasks. FP-Growth and BPFP is estimates work load of every mining unit and it divides so many groups than balance load among every group [3].

FP-growth algorithm is a relatively tree structure. The multi scan problem is avoided and also successfully avoids the generation of candidate item set at every iteration. This algorithm consists of two full data set scan, also crates frequent patterns tree and speed has improved dramatically [1].

FP-growth adopts two ways those are divide and conquer way. Grided FP-Growth, GFP-Growth is short and it is designed to run on computer cluster. To avoid memory overflow, GFP-Growth adopts the projection method to find all the conditional pattern bases directly without constructing an FP-tree [2].

Pervasive and grid computing large scale computation power, complex problems and also very large data storage resources. Grid computing is a loosely coupled low cost architecture based on internet connection. It is heterogeneous computing and storage is traditional cluster system. Grid computing is very easy and also add some additional computing resources [4].

In big data analytics one of the fundamental problem is FPM (Frequent Pattern Mining). It works on transaction. Every transaction has a number of non-repeated items such as associates and desecrate entities etc., FPM has a major application in analytics such as transportation industry, telecom, retail, healthcare and finance etc. [5].

The spark is a open source distributed framework designed by the Brekely Lab. Which can applied for Machine Learning, Data Mining and any iterative algorithms [7].

At present there are many iterative incremental algorithm for mining association rules like hashing and pruning, fast update based on the Apriori algorithm. This algorithms used to though threshold values lower, upper to define the item frequency. This kind of algorithm doesn't need any rescan of items [8].

II. RELATED WORK

In paper [1] author proposed a new algorithm is called PFP-growth (Parallel FP- growth). This algorithm proposed parallel frequent item set mining.

The mining strategies divides different stages that is to achieve balance between processor and some data structure is adopt to reduce the information transportation between processors.

In paper [2] author proposed a new algorithm is called novel parallel FP-Growth. It is designed to run on the computer cluster and also avoid memory overflow.

This algorithm shows all conditional pattern bases of frequent items by projection without any construction on FP-tree.

In paper [3] author proposed an algorithm known as balanced parallel FP-Growth algorithm BPFP based on PFP algorithm. It parallelizes the MapReduce in the FP-Growth. The BPFP added into the PFP-Growth load balance feature. It improves performance and also parallelization.

In paper [4] author discussed a new algorithm is called as BTP-tree (Balanced Tide set Parallel FP-tree). It is efficient parallel and distributed mining algorithm on grid computing and it is also heterogeneous computing environment. BTP- tree is used to find tree width and depth.

In paper [5] author introduce parallel child MapReduce. It is a one of the programming model and it creates dynamically and synchronized in a hierarchical parent child fashion. Using MapReduce Parent-child MapReduce has been used for parallelize recursive divide and conquer algorithm. By using this model can lead speedups significant in the computational speed.

In paper [6] author discussed new algorithm called as DFPS. It is partitioned in different computing tasks in each computing build. Conditional FP-tree and adopts a pattern fragment growth method to mine the frequent item sets independently. Distributed functional point algorithm based on spark does not pass messages between nodes during frequent item set and frequent mining.

In paper [7] author presents a distributed FP-Growth algorithm by using the spark. The results proved that compare the MapReduce on spark. This algorithm is high flexible and efficiency.

In [8] author proposed a Pre-FP based on large item sets, which doesn't need the rescan of the database. It achieves a good efficiency for construction of the FP-tree.

In [9] author proposed a Novel Parallel Pattern Tree structure (PP-tree) that reduces the input output cost by performing the single scan to extract the contents of a database in efficient way. This algorithm will work independently at both locally and globally sites.

III. PROBLEM DEFINITION

Here in the distributed databases the data was divided into 2 parts. Let the transaction and the items be T and $(t_1, t_2, ..., t_n)$. The transaction id is represented as TID. $X \Rightarrow Y$ is an association rule, where $X \subset T$, $Y \subset T$ and $X \cap Y = \varnothing$. If $C\%$ of transaction is X and Y, $X \Rightarrow Y$ are with confidence C. $X \Rightarrow X$ is with support S. If $S\%$ of transaction contain $X \cup Y$. [14]

In mining Boolean association rules 0 or 1 represents the present obscene of an attributes. The transactions also may be a string of 1 or 0 then simply counts the attributes which consists of 1. It can also define as the attribute number is 1+a where Z is with b-attributes $Z_1, ..., Z_b$ and D is with attributes $D_1, ..., D_a$. There are 1+a 0 or 1 sequence where transaction number is n. $\vec{X}$ and $\vec{Y}$ data base i.e., $m_i = 1$ iff row i is 1 for attribute $\vec{X}$. The product of $\vec{X}$ and $\vec{Y}$ is

$$\vec{X} \times \vec{Y} = \sum_{i=1}^{n} g_i h_i \quad ....(1)$$

Whether $\vec{X}, \vec{Y} \geq K$ means whether (XY) is frequent. The algorithm to compute $\vec{X}.\vec{Y}$ is address in the current paper. To identify the frequent one item set, A with 2 attributes and B with 3 attributes, need to know whether the item set $b = A_g, A_h, B_g, B_{h,}, B_c$ is frequent. A creates the vector $\vec{X}$ where $\vec{X} = \vec{A_g} \times \vec{A_h}$ (component multiplication) and B creates the vector $\vec{Y}$ where $\vec{Y} = \vec{B_g} \times \vec{B_h} \times \vec{B_c}$ product of $\vec{X}$ and $\vec{Y}$ generates the result item set. All association rules is completed by using support is greater than or equal to minimum support.[14]

The privacy preserving without reliving the individual transaction is explained in next section.

IV. IV. PRIVACY PRESERVING ALGORITHM FOR DISTRIBUTED SERVERS

In the proposed algorithm the values in the equations is marked with random values for computation of the scalar product. Where $R_1, ...R_n$ is the random values, n is not an odd.

$\vec{X}$ is generated and Q are coefficient for the equation server 1 sends X_g to server 2.

$$[\mathrm{g}_1 + \mathrm{q}_{1,1} \times \mathrm{R}_1 + ...\mathrm{q}_{1,\mathrm{n}} \times R_{\mathrm{n}}]$$

$$[\mathrm{g}_2 + \mathrm{q}_{2,1} \times \mathrm{R}_1 + ...\mathrm{q}_{2,\mathrm{n}} \times R_{\mathrm{n}}]$$

.

.

.

$$[\mathrm{g}_\mathrm{n} + \mathrm{q}_{\mathrm{n},1} \times \mathrm{R}_1 + ...\mathrm{q}_{\mathrm{n},\mathrm{n}} \times R_{\mathrm{n}}] \quad ....(2)$$

Then server 2 calculates $\vec{g}$. $\vec{h}$ and n values

$$[\mathrm{q}_{1,1} \times \mathrm{h}_1 + ...\mathrm{q}_{1,\mathrm{n}} \times \mathrm{h}_\mathrm{n}]$$

$$[\mathrm{q}_{2,1} \times \mathrm{h}_1 + ...\mathrm{q}_{2,\mathrm{n}} \times \mathrm{h}_\mathrm{n}]$$

.

.

.

International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

$$[q_{n,1}\times h_1+\ldots q_{n,n}\times h_n] \ldots\ldots\ldots\ldots\ldots(3)$$

Due to the n independent equations in $(h_1,\ldots h_n)$ in the sever 1 it cannot deliver them to the server 2. It reviles the values of b. So, the server 2 will generates r random values $R'_1\ldots R'_r$. The X value is governed by r and the n values are partitioned by the server to as below.

$$\begin{gathered}
[q_{1,1}\times h_1+\ldots+q_{1,n}\times h_n+R'_1] \\
\vdots \\
[q_{1,n/j}\times h_1+\ldots+q_{1,n/j}\times h_n+R'_1] \\
[q_{1,n/j+1}\times h_1+\ldots+q_{1,n/j+1}\times h_n+R'_2] \\
\vdots \\
[q_{1,2n/j+1}\times h_1+\ldots+q_{1,2n/j+1}\times h_n+R'_2] \\
\vdots \\
[q_{1,(R-1)n/r+1}\times h_1+\ldots+q_{1,(R-1)n/j+1}\times h_n+R'_R]. \\
\vdots \\
[q_{1,n}\times h_1+\ldots+q_{1,n}\times h_n+R'_R] \ldots\ldots(4)
\end{gathered}$$

J and the n above values are send to the server 1 by server2.

$$\begin{aligned}
J &= (g_1+q_{1,1}\times R_1+\ldots q_{1,n}\times R_n)\times h_1 \\
&+(g_2+q_{2,1}\times R_1+\ldots q_{2.n}\times R_n)\times h_2 \\
&\vdots \\
&+(g_n+q_{n,1}\times R_1+\ldots q_{n.n}\times R_n)\times h_n \ldots\ldots\ldots\ldots(5)
\end{aligned}$$

By doing grouping of $g_i\times h_i$

$$\begin{aligned}
J &= (g_1\times h_1+\ldots+g_n\times h_n) \\
&+(h_1\times q_{1,1}\times R_1+\ldots+h_n\times q_{1,n}\times R_n) \\
&+(h_1\times q_{2,1}\times R_1+\ldots+h_n\times q_{2,n}\times R_n) \\
&\vdots \\
&+(h_1\times q_{n,1}\times R_1+\ldots+h_n\times q_{n,n}\times R_n) \ldots\ldots(6)
\end{aligned}$$

It is $\sum_{i=1}^{n} g_i\times h_i$. To make a change and factor out R_i, we rearrange the above equation

$$\begin{aligned}
J &= \sum_{i=1}^{n} g_i\times h_i \\
&+R_1\times(h_1\times q_{1,1}+\ldots+h_n\times q_{n,1}) \\
&+R_2\times(h_1\times q_{1,2}+\ldots+h_n\times q_{n,2}) \\
&\vdots \\
&+R_n\times(h_1\times q_{1,n}+\ldots+h_n\times q_{n,n}) \ldots\ldots\ldots(7)
\end{aligned}$$

The above equation can be rewrite by adding and subtracting the same quantity from one side:

$$\begin{aligned}
J &= \sum_{i=1}^{n} g_i\times h_i+\{R_1\times q_{1,1}\times h_1+\ldots+q_{n,1}\times h_n)+R_1\times R'_1-R_1\times R'_1\} \\
&\vdots \\
&+\{R_{n/j}\times(q_{1,n/j}\times h_1+\ldots+q_{n,n/j}\times h_n)+R_1\times R'_1-R_1\times R'_1\} \\
&+\{R_{n/j+1}\times(q_{1,n/j+1}\times h_1+\ldots+q_{n,n/j+1}\times h_n)+R_2\times R'_2-R_2\times R'_2\} \\
&\vdots \\
&+\{R_{2n/j}\times(q_{1,2n/j+1}\times h_1+\ldots+q_{n,2n/j+1}\times h_n)+R_2\times R'_2-R_2\times R'_2\} \\
&\vdots \\
&+\{R_{2n/j+1}\times(q_{1,2n/j+1}\times h_1+\ldots+q_{n,2n/j+1}\times h_n)+R_3\times R'_3-R_3\times R'_3\} \\
&\vdots \\
&+\{R_{(R-1)n/j}\times(q_{1,(R-1)n/j+1}\times h_1+\ldots+q_{n,(R-1)n/j}\times h_n)+R_{R-1}\times R'_{R-1}-R_{R-1}\times R'_{R-1}\} \\
&+\{R_{(R-1)/R+1}\times(q_{1,(R-1)n/j+1}\times h_1+\ldots+q_{n,(R-1)n/j+1}\times h_n)+R_R\times R'_R-R_R\times R'_R\} \\
&\vdots \\
&+\{R_n\times(q_{1,n}\times h_1+\ldots+q_{n,n}\times h_n)+R'_R-R'_R \\
&\ldots\ldots\ldots\ldots\ldots(8)
\end{aligned}$$

Server 1 factors out R_i from above equation

$$\begin{aligned}
J &= \sum_{i=1}^{n} g_i\times h_i+R_1\times(q_{1,1}\times h_1+\ldots+q_{n,1}\times h_n+R'_1) \\
&\vdots \\
&+R_{n/j}\times(q_{1,n/j}\times h_1+\ldots+q_{n,n/j}\times h_n+R'_1) \\
&+R_{n/j+1}\times(q_{1,n/j+1}\times h_1+\ldots+q_{n,n/j+1}\times h_n+R'_2) \\
&\vdots
\end{aligned}$$

$$+R_{2n/j}\times(q_{1,2n/j+1}\times h_1+\ldots+q_{n,2n/j+1}\times h_n+R_2')$$
$$+R_{2n/j+1}\times(q_{1,2n/j+1}\times h_1+\ldots+q_{n,2n/j+1}\times h_n+R_3')$$
$$\vdots$$
$$+R_{(R-1)n/j}\times(q_{1,(R-1)n/j}\times h_1+\ldots+q_{n,(R-1)n/j}\times h_n+R_{R-1}')$$
$$+R_{(R-1)/R+1}\times(q_{1,(R-1)n/j+1}\times h_1+\ldots+q_{n,(R-1)n/j+1}\times h_n+R_R')$$
$$\vdots$$
$$+R_n\times(q_{1,n}\times h_1+\ldots+q_{n,n}\times h_n+R_R')$$
$$-R_1\times R_1'-R_2\times R_1'-R_{n/j}\times R_1'-R_{n/(r+1)}\times R_2'-R_{n/r+2}\times R_2'-R_{2n/r}\times R_2'$$
$$\vdots$$
$$-R_{(r-1)n/r+1}\times R_r'-R_{(r-1)n/(r+2)}\times R_r'-R_n\times R_r' \quad \ldots\ldots\ldots\ldots\ldots\ldots(9)$$

Server 1 already has R_i. Server 2 deliver the coefficients of R_i. Server 1 multiplies the n values with R_i

$$T=\sum_{i=1}^{n} g_i\times h_i$$
$$-R_1\times R_1'-R_2\times R_1'-R_{n/j}\times R_1'-R_{n/j+1}\times R_2'-R_{n/j+2}\times R_2'-R_{2n/r}\times R_2'$$
$$\vdots$$
$$-R_{(j-1)n/j+1}\times R_j'-R_{(j-1)n/j+2}\times R_r'-R_n\times R_j' \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(10)$$

Do factoring with R_j'

$$T=\sum_{i=1}^{n} g_i\times h_i$$
$$-(R_1+\ldots+R_{n/j})\times R_1'-(R_{n/j+1}+\ldots+R_{2n/j})\times R_2'$$
$$\vdots$$
$$-R_{(j-1)n/j+1}+\ldots+j_n)\times R_j' \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(11)$$

We should calculate $T=\sum_{i=1}^{n} g_i\times h_i$, server 1 needs to add the r values sent from server 1 to server 2, and server 2 is finally with the result.

V. FP-GROWTH ALGORITHM ON HADOOP

In the scan of the server by using the FP-Growth is deriving the set of frequent 1-item set with the support count. Here for example the support count be 2 the frequent item sets we sorted in descending support count order then the resultant is denoted as F={(f2:7), (f1:6),(f3:6), (f4:2), (f5:2)} then the frequent pattern tree is constructed as follows.

The route is created for the tree and labeled as null. The server is scanned second time and all the transaction is processed and arranged the order F then the branch was created every transaction after constructing the whole tree it look like the figure 1.

T = T1; T2; ; Titems is the set of item. The noise set is R = R1;R2; ;R_{dummy}. The set of prime number P = P1; P2; ; Pn. If Pi satisfies Eq.(12)(13)(14), Pi is the key of Ti. Dummy item is generated as shown in Fig 1: Dummy item is generated by service provider as noise and assigns prime numbers to them. A table of is maintained.

Here the key used because to distinguish the dummy item and the original items.

$$\text{key}(I_i)=P_i \quad (1\leq i\leq \text{items})\ldots\ldots\ldots\ldots\ldots(12)$$

$$\text{key}(R_j)=P_{items+j} \quad (1\leq j)\ldots\ldots\ldots\ldots\ldots\ldots(13)$$

$$P_i\neq P_j \;;(\forall i\neq j \text{ and } P_i,\ P_j\in P)\ldots\ldots\ldots\ldots\ldots(14)$$

FP-Growth is executed on Hadoop platform. The executed process is shown in figure 1. Each transaction has the item with the key there is used in the MapReduce and the prime numbers that mentioned is above is also used in the MapReduce.The key is as

$$\text{Key}(F')=\prod_{t=1}^{m} key(F_{r_t})=\prod_{t=1}^{m} P_{r_t} \quad \ldots\ldots\ldots\ldots\ldots\ldots(15)$$

Here the key is removed dummy items.

FID	List of items IDS
F100	(I1,3),(I2,4), (I5,8)
F200	(I2,6),(I4,7)
F300	(I2,5),(I3,6)
F400	(I1,9),(I2,8),(I4,3)
F500	(I1,5),(I3,9)
F600	(I2,6),(I3,4)
F700	(I1,5),(I3,7)
F800	(I1,5),(I2,4), (I3,9),(I5,6)
F900	(I1,9),(I2,8),(I3,6)

Dummy	Key
I1	17
I2	13,23
I3	19

Original	Key
I1	2
I2	7
I3	3
I4	5
I5	11

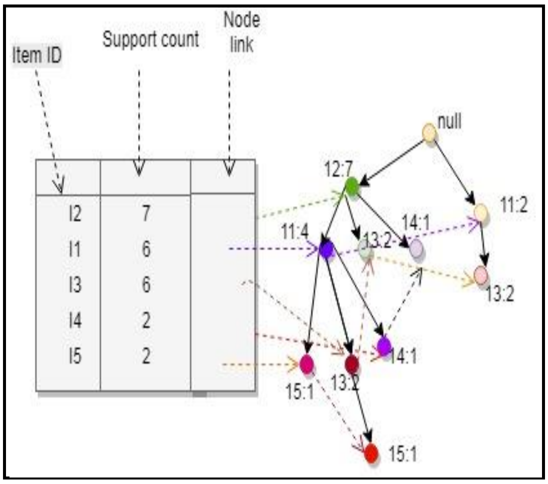

Figure 1: FP-Growth on Hadoop

Table 1: Security Analysis of algorithm on distributed servers

	Values	Random Number Num	Not known of number	Equation Num
g	$g_1 \ldots g_n$	n	2n	n+J
h	$h_1 \ldots h_n$	J	n+J	n

VI. EXPERIMENT

Here the analysis of performance and the security proposed algorithm was carried out. The experimental analysis proved that the proposed algorithm will not increase the cost and the communications by avoiding the rescans of the database and in addition to that the private information was reserved.

a. Privacy preserving distributed server algorithm:

By adding the random numbers in the values of the equation the algorithm secured the privacy of the transactions. The random number was chosen privately and randomly. The random number can be calculated by taking the enough information for another server. So with the analysis as in the table 1 and the mathematical proof in the section IV proved that these method achieve that privacy preserving.

b. An algorithm on Hadoop:

Hadoop cluster is used in this experiment. Several nodes are there in clusters such as windows server 2012 64 bit and name node are 58-core 12 GHz, 120 GB memory. Algorithm was described in section V and figure 1. In section V, we have established evidence that our method is privacy preserving. Now we estimate the time cost because it is most important than the other factor. Then we did more different practical analysis.

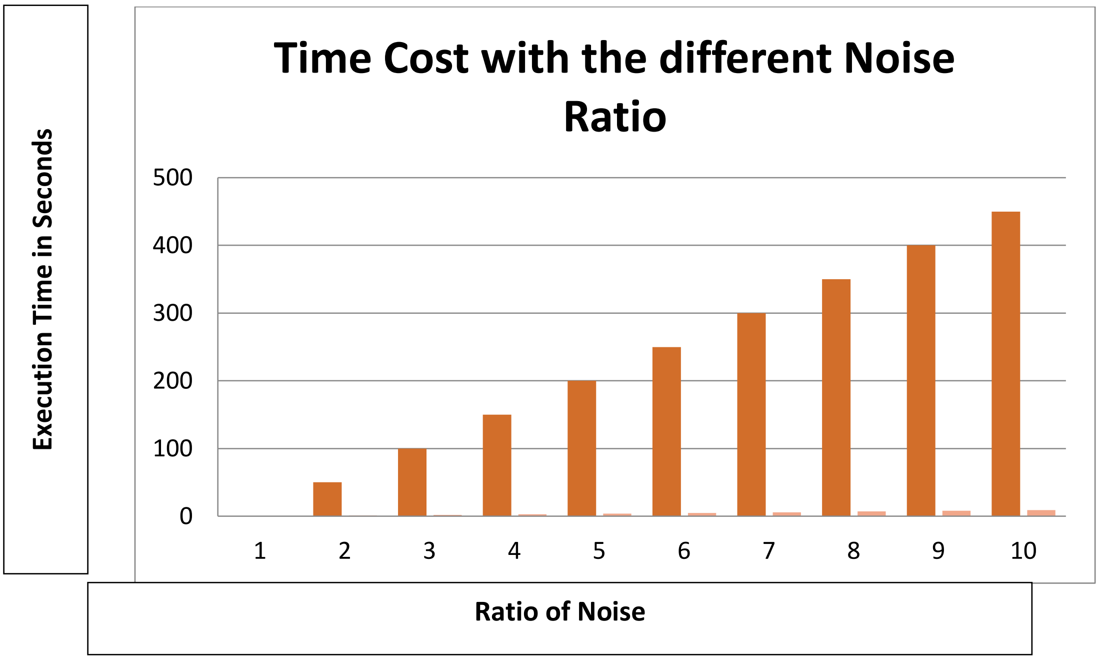

Figure 2: Time Cost with the Different Noise Ratio

System performance is decreased with the increase of noise. The performance is survey by differentiate noise ratio. Dummy type number is 25, transaction number is 500, dummy type is 10, noise is from 0% to 40%, in addition is the support value is 50% as shown in Figure 2. The execution noise size and execution time are same but not significantly. So the new technique does not significant influence the performance.In fig 3, it is solved in various sizes of transactions. Based on dummy type number, support value and the type of item was same as before. The size of noise is 25% compared with 55% and sizes of transactions are increased from 300 to 700. It is display the execution time related to the transaction data size.

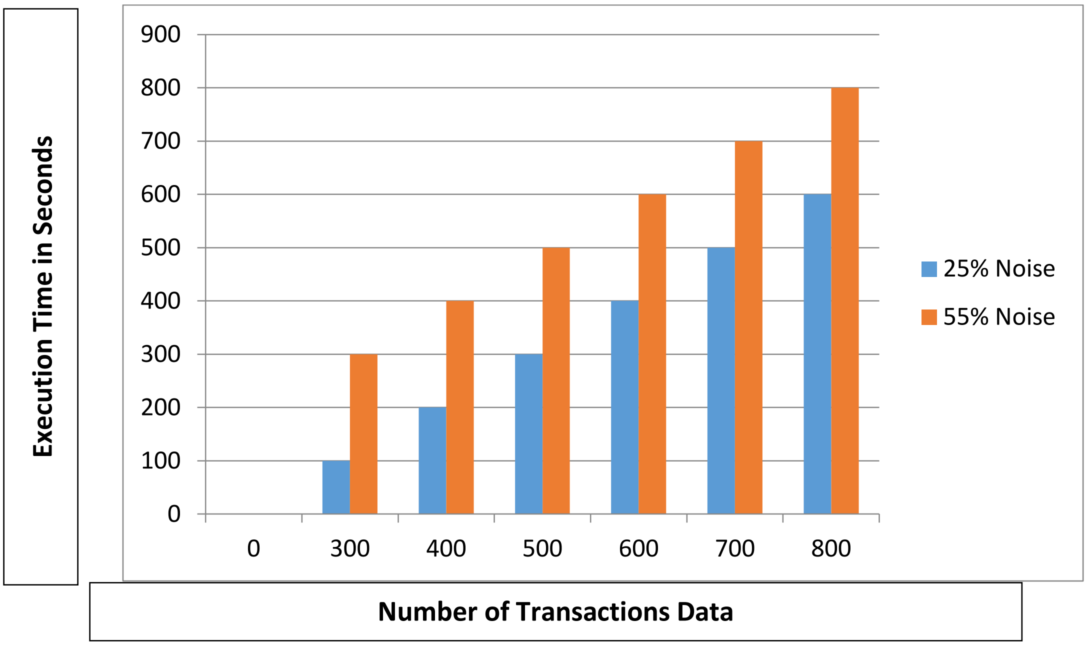

Figure 3: Cost in various sizes of transactions.

Evaluated of system performance is different transaction length. Transaction length and time cost is related. In figure 4 Shows dummy item is added and the number dummy items is increased. Some other experimental parameters are also as the pervious experiments. The size of transaction is 500. The time cost increases as well as transaction length is also increased. All parameters are analyzed based on above affects the time cost. Whenever, it is not an important issue.

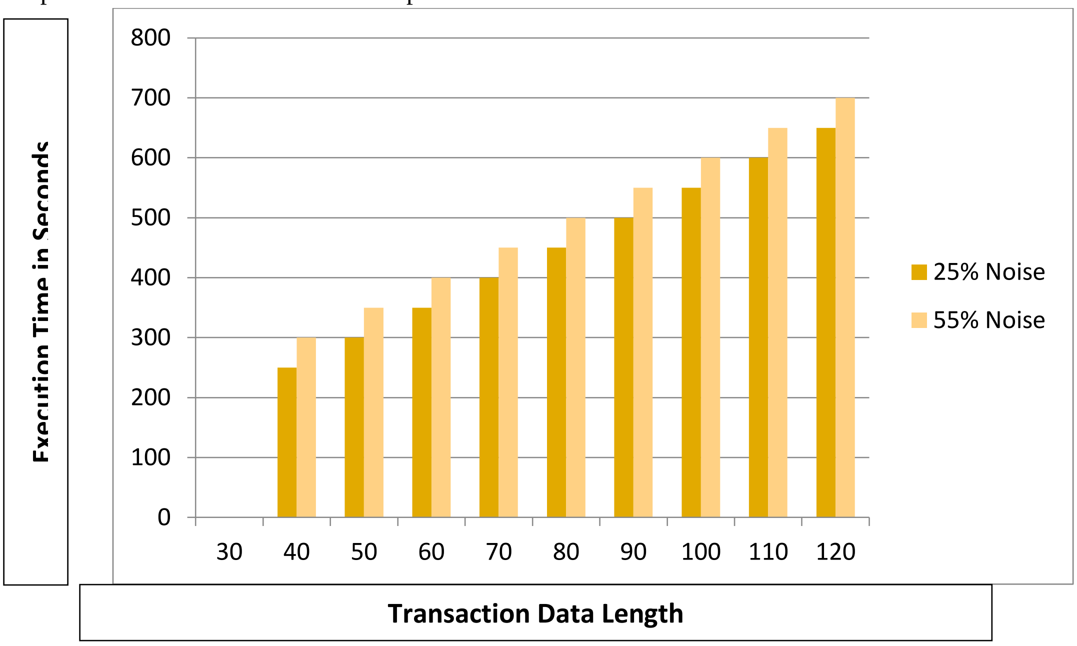

Figure 4: Cost changes with respect to the transaction length

The picture of time cost with various those are 'length of transaction data, number of transaction data, ratio of noise, and plot a 3 different dimensional figures the color that means time cost.

Finally the conclusion, we create a comparison by increasing the length of transaction data, increasing noise size and number of transaction data. The experimental results shows time cost is not influenced and that privacy can be protected.

VII. CONCLUSION

Here mining association rules algorithm with preserving the data of a transaction was designed for the distributed servers in a cloud by using the corresponding protocols. It uses the method of machine learning in addition with the privacy preserving for the distributed servers. Similarly in the Hadoop platform by applying the dummy data i.e. noise to the transactional data in the public clouds, the results shows that the private data is secured and at the same time the cost of the execution is not significant by mining the association rules.

REFERENCES

1. Li, Li, Donghai Zhai, and Fan Jin. "A parallel algorithm for frequent itemset mining." *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*. IEEE, 2003.
2. Chen, Min, Xuedong Gao, and HuiFei Li. "An efficient parallel FP-Growth algorithm." *Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009. CyberC'09. International Conference on*. IEEE, 2009.
3. Zhou, Le, et al. "Balanced parallel fp-growth with mapreduce." *Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on*. IEEE, 2010.
4. Zhou, Jiayi, and Kun-Ming Yu. "Tidset-based parallel FP-tree algorithm for the frequent pattern mining problem on PC clusters." *International Conference on Grid and Pervasive Computing*. Springer, Berlin, Heidelberg, 2008.
5. Makanju, Adetokunbo, et al. "Deep parallelization of parallel FP-growth using parent-child MapReduce." *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016.
6. Shi, Xiujin, Shaozong Chen, and Hui Yang. "DFPS: Distributed FP-growth Algorithm Based on Spark."
7. Deng, Lingling, and Yuansheng Lou. "Improvement and research of fp-growth algorithm based on distributed spark." *Cloud Computing and Big Data (CCBD), 2015 International Conference on*. IEEE, 2015.
8. Jian-Ping, Liu, Wang Ying, and Yang Fan-ding. "Incremental mining alogorithm Pre-FP in association rules based on FP-tree." *Networking and Distributed Computing (ICNDC), 2010 First International Conference on*. IEEE, 2010.
9. Tanbeer, Syed Khairuzzaman, Chowdhury Farhan Ahmed, and Byeong-Soo Jeong. "Parallel and distributed frequent pattern mining in large databases." *High Performance Computing and Communications, 2009. HPCC'09. 11th IEEE International Conference on*. IEEE, 2009.
10. A. Javed and A. Khokhar, "Frequent Pattern Mining on Message Passing Multiprocessor Systems," *Distributed and Parallel Databases*, vol. 16, 2004, pp. 321–334.
11. J. Hu and X. Yang-Li, "A Fast Parallel Association Rules Mining Algorithm Based on FP-Forest," *Proc. 5th Int. Symposium on Neural Networks*, 2008, pp. 40–49.
12. K.-M. Yu, J. Zhou, and W.C. Hsiao, "Load Balancing Approach Parallel Algorithm for Frequent Pattern Mining," *Proc. PaCT*, 2007, pp. 623–631.
13. O.R. Zaane, M. El-Hajj, and P. Lu, "Fast parallel association rule mining without candidacy generation,"*Proc. ICDM*, 2001, pp. 665-668.
14. Huang, Kai, et al. "Distributed Apriori on cloud plateform without exposing individual transaction data." *Advanced Computational Intelligence (ICACI), 2018 Tenth International Conference on*. IEEE, 2018.

AUTHORS PROFILE

Ms Saritha Byreddi, Research Scholar, Department of Computer Science and Engineering, Sri Venkateswara Univrsity, Tirupati. Ms Saritha Byreddi, is Working on Distributed Data Mining at Sri Venkateswara University, Tirupati. She pursuing her PhD in the area of Distributed Data Mining and her research Contributes are to extract the association rules in the distributed environment. She completed her Bachelors (B.Tech) & Master's (M.Tech) Degree in the Specialization of Computer Science and Engineering at JNTU. Shee has more than 10 years teaching experience from various Institutions. She has attended more than 10 Workshops specifically in the area of DDM. He has published 5+ National and International Conferences and Journals.

Dr.A.Rama Mohan Reddy He **r**eceived his B.Tech Degree from JNT University Anantapur in 1986, Masters in Computer Science and Engineering from NIT, Warangal in 1991 and Ph.D. in Computer Science and Engineering from Sri Venkateswara University, Tirupati in 2007. He is currently working as a Professor of Computer Science and Engineering, SV University College of Engineering, Tirupati, India. His research interests are Software Engineering, Software Architecture, Cloud Computing, Operating System and Data Mining. He is life member of ISTE, IETE, ISC and CSI. He has more than 30 years of experience in teaching, 7 scholars completed Phd.'s under his guidance. He has 100+ publications and presented 78+ papers in National and International conferences.