A New Pattern Mining Algorithm for Analytics of Real-Time Internet of Things Data

Monika Saxena, C.K. Jha, Deepika Saxena

Abstract: ***The rise of IoT Real time data has led to new demands for mining systems to learn complex models with millions to billions of parameters, which promise adequate capacity to digest massive datasets and offer powerful predictive analytics. To support Big Data mining, high-performance powerful computing platforms are required, which impose regular designs to unleash the full power of the Big Data. Pattern mining poses a lot of interesting research problems and there are many areas that are still not well understood. The specifically very elementary challenges are to understand the meaningful data from the junk data that anticipated into the internet, refer as "Smart Data". Eighty-five percent of the entire data are noisy or meaningless. It is a very tough often assigned to verify and separate to refine the data from the noisy junk. Researchers' are proposing an algorithm of distributed pattern mining to give some sort of solution of the heterogeneity, scaling and hidden Big Data problems. The algorithm has evaluated in parameters like cost, speed, space and overhead. Researchers' used IoT as the source of Big Data that generates heterogeneous Big Data. In this paper, we are representing the results of all tests proved that; the new method gives accurate results and valid outputs based on verifying them with the results of the other valid methods. Also, the results show that, the new method can handle the big datasets and decides the frequent pattern and produces the associate rule sets faster than that of the conventional methods and less amount of memory storage for processing. Overall the new method has a challenging performance as regard the memory storage and the speed of processing as compared to the conventional methods of frequent pattern mining like Apriori and FP-Growth techniques.***

Keywords: Internet of Things, Pattern Mining, Real time analytics, Big data

I. INTRODUCTION

The most important questions arise at this point is that, how these immense amounts of big data can be stored and be processed? How to fetch the meaningful data from millions of millions of records of data? In order to answer this question, it should be known that, not all pieces of the big data are important; a lot of them are redundant information and some are valueless. Consequently, filtering these amounts of database on its weight of meaningfulness value is the primary clue for solving this problem. Consequently, it will be possible to distinguish the unique information and filter the meaningful data, and in turn, it saves storage and processing time. On the other hand, by determining the frequent patterns of data, it could help greatly to predict the associate rule sets that can be taken as a guide in deducing the behavior of systems in advance based on the historical data. This approach is called data and frequent pattern mining.

Distributed pattern mining is one of the solution methods to improve the performance of processing the Big Data. Furthermore, it saves Exabyte of storage space alongside with saving the processing time. Not only that but also, it widely opens the door for mining thousands of rule sets that are used in predicting facts and reveal the mysterious behavior of the unknown systems[3]. Data mining process is deployed by running some parallel programming tools like SAMOA (Scalable Advanced Massive Online Analysis) or Map Reduce [4].

A. Internet of Things

The concept of IoT was defined by a member of the RFID (Radio Frequency Identification) development community in 1999, and it recently become relevant to the world because of the growth of mobile devices, sensors, ubiquitous communication, cloud computing and data analytics[1]. In this millions and billions of objects can communicate share and sense information. Internet of things is a physical network of sensors, the sensor can be inbuilt in any of the physical device. The communication and information sharing will be done by different number of protocols.

The Internet of Things is a variable for designers because end point "things" are the main factor. There are some characteristics of IoT have found. Each characteristics having it own functionality, we can say here the IoT is multidisciplinary. We can describe it on five key characteristics of IoT they are following.

Intelligence

Algorithms and computation makes device intelligence, a product or thing is called smart when it has capability to decide and compute different operations. It includes actionable intelligence and decision making capability.

Connectivity

Connectivity of different things can be done by Wi-max, Bluetooth etc. any other wireless communication architecture. It enables network accessibility and compatibility in terms of produce and consumes data.

Sensing

It is the ability to understand, physical world and environment. Basically it takes data as in analog input and convey to device, which is responsible to processing.

Expressing

It is the ability to express processed data into physical world and environment. Basically it gives output data as in analog and convey to physical world, which is called processed data.

Energy

As, all devices requires energy to process. So this is the characteristics of IoT, because power consumption and maintainace is the biggest issue or research area now-a-days.

Revised Manuscript Received on November 06, 2019.

***** Correspondence Author

Dr. Monika Saxena*, Assistant Professor, Computer Science Department, Banasthali Vidyapith, Banasthali, India. Email:muskan.saxena@gmail.com

Prof. C .K. Jha, Professor, Head, Computer Science Department, Banasthali Vidyapith, Banasthali, India. Email:ckjha1@gmail.com

Ms. Deepika Saxena, Computer Science Department, Sarvepalli Radhakrishnan University, Bhopal, India. Email:deepika.mist@gmail.com

We define IOT into Big Data, and Big Data defines the enormous amount of data that can be digitized, which will collect from different sources. Data will oversized and acutely in the form. Large amount of data gets a handle on new and different challenges like complexities, securities, risks to the privacy of data [5]. Actually, we need to fetch 'Smart data' from the Big complex Data.

Dynamic Big data defines a variety of data. For example, in social sites user uses many files like images, videos, text, etc., in which some are useful data, and some are not useful. There are two types of dynamic data:

Homogeneous- Data set is composed of the same set of features over distributed data sets.

Heterogeneous- Data set is composed of the different set of features over distributed data sets [3]

Management of large amount data is naturally dissimilar from conventional relational models of data management. Structural data have easily managed by relational modeling of data, but unstructured data will manage in a different way [4] [5]

II. RESEARCH OBJECTIVES

The few lines introduced above, defines the problem which is the point of this research as how to filter the meaningful data or the smart data from the Internet of Things Real time data and mining the frequent patterns and in turns predict the associated rules. The main goal of this research is to propose a new method for mining the frequent pattern from a Big having the space of storage and the time of processing are optimized. In order to realize this aim, there are many objectives should be done by this research.

These objectives are surveying the literature for investigating the current techniques of distributed frequent pattern mining like for example Apriori and FP-Growth including the data structures are used, algorithms, strong and weak points of each technique. Thereby, the new proposed technique avoids the drawbacks of the conventional techniques and keeps on the good features alongside adding an improvement in terms of reducing the time of processing and or reducing the space of storage and enhancing the performance of Big Data in parameters like communication overhead, computational costs, memory usage and throughput.

Furthermore, in order to test the new method, Big Data sets are needed to run the test cases on it and validate the successfulness and the proper operation of the new algorithm and also to compare its results with the results of the traditional methods. So, a hundreds of thousands of records of data were collected from different data sources in different aspects like Smart Cities data sets including parking , transportation , pollution and Real Time IoT dataset and many process were run for data preparation and preprocessing prior to the main process of the test of the proposed algorithm.

III. RESEARCH APPROACH

In this research researchers proposing a new distributed pattern mining algorithm to overcome the performance or the space of storage related problems in parameters like minimizing the number of candidates set and minimizing the number of paths of scanning the database transactions which in turn reducing the communication overheads, throughput, memory usage & computational costs of I/O of Big Data Mining. The new method will be categorized as Apriori-like method. However, it is completely difference in searching techniques and forming the candidates and item set lists. The proposed method can complete the process efficiently in just one path of scan of the database transactions with a significant reduction in the number of frequent sets and in turn reducing the number of tests for the frequent of the pattern set. Moreover, the only one path of scan, applies the binary search technique which has a complexity of N LOG N while the traditional Aproiri uses many paths of scan applying the linear search algorithm with complexity of N^2 . Furthermore, it will save the required space for a tree structure and its nodes and linking pointers which is the drawback of the FP-Growth method.

A. Proposed Framework

The agent works for both distributed and decentralized data fusion. Meanwhile, proposed algorithm is used to categorize the patterns and information with low latency**.**

Fig. 1 representing the whole process done by researcher. The process is described following,

- Firstly, Data has collected from different sources in the form of static and dynamic both. The different Big Data resources are used mainly are Smart City: Transportation: parking, Traffic Data Sets, Pollution Data Set and Internet of Things Data Set.
- The collected data has cleaned using algorithm and some defined methods. Dynamic data cleaning is done by different cleaning method.
- Finally, proposed algorithm has applied on each and every nodes or agents to get patterns. After that the all different patterns are integrated using local model aggregation and Data fusion method. After that final model is ready to access actual required data with low latency and low overhead.

B. The Proposed Method

The new method has been designed to remove the limitations of the Apriori algorithm considering keeping the good feature like easy implementation, and moderate memory storage requirements. Furthermore, fixing the drawbacks like saving the time of and the number of scanning passes is reduced to be only one pass. Not only are the objectives of the new technique to keep the good features, avoid the drawbacks, and fixing the disadvantages. The new method aims to utilizing the multilevel hashing method based on transaction weight in searching the database transactions throughout the hundreds of thousands of transactions instead of linearly searching the database to reduce the time and accelerate the whole process.

Starting from the point that, the data has been processed and ready for processing including the support calculation and compare it to the smallest support to decide which item will be considered in forming the item set and which will be discarded due to it is lower than the minimum support. The method works in one and only one pass , starting by reading the transaction and generate the possible combinations and store in the form of multilevel hash table of items in pairs , triple , quadruple , and so force based on the length of the transaction. Then it search the item set pool to check if it is a new item set or and existent item set. If it exists, then it just

increments the frequent value corresponding to this item set. And if it is a novel item set, it then creates the item set and add it to the pool and assigns an initial value of 1 as its frequent. And then it moves to the next transaction for processing and repeat the process until it processed all transactions of the dataset.

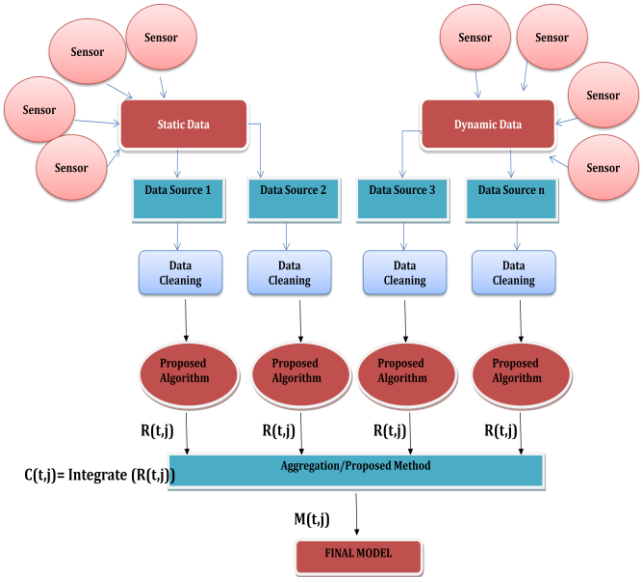

Fig. 1 Proposed Framework

C. Proposed algorithm in distributed Framework

Big Data (e.g. sensors) are collected from specific to domains. Resources are distributed with the [19], fault-tolerance, strict security requirements and agility. Collected data need to be combined. For final modeling data will combined in a manner , the process to combine different domain data is called “Data fusion”.

This is the process of combing data from different sources. It provides a robust process of interest. Data fusion required where analytic algorithms that can combine the results from distributed systems. This is often divided into four processes. In Level 1 and 2 data will concerned with processing using numerical fusion methods such as Kalman filtering or probability theory. Other levels 3 and 4 are concerned with the extraction of knowledge from level 1 and 2, It is basically contains decision making system.

In this context, Researcher using recursive data fusion methodology as follows:

• Agent j represents a sensor and it does not communicate with any other sensors. but it works with its peer sensor. The list of peer specified by the agents.

• Agent j includes an engine of learning method that analyzes and collects data from its defined knowledge base R(t,j), it is the representation of bi-gram feature.

• Agent j includes a fusion engine with proposed algorithm hat can be customized externally. proposed algorithm integrates the all local knowledge base R(t,j) and passed along to its peers, it uses recursion. As shown in Figure 4.6. proposed algorithm assesses the total value of agent j by separating the total knowledge base into the categories of patterns, emerging and anomalous

Below two steps recursively follow until nth source or agent

Step 1: R(t,j) = function1(C(t-1, k(j)), R(t,j));

Step 2: M(t,j) = function2(C(t,j))

Step3: T[hi(M(t,j))]

Where k(j) represents the peer list of agent j

T[hi(M(t,j)] represents multilevel hashing of knowledge base.

Function 1 integrates the local knowledge base R(t,j) to the total knowledge base C(t,j) that can be passed along to its peers and used globally in the recursion .

Function 2 assesses the total value of the agent j by separating the total knowledge base into the categories of patterns, emerging and anomalous themes based on the total knowledge base C(t,j) and generates M(t,j).

IV. Simulation and Results

The programming work has been done by using MATLAB R2016a and C++ language. The first implementation version of the work has been done by using C++, three versions, one for Apriori, one for FP-Work and one for the new proposed technique. All these versions of work have been confirmed with the valid Java Open-Source Data Mining Library (SPMF) version 2.19 to make sure of the correctness of the developed work. Proposed solution is simulated with new modified algorithm, test deployment of now able to support Scalability and heterogeneity of data and traffic control.

A. Test case: Internet of Things Data Set

In smart cities, hundreds of thousands of sensors are working all the time to log the physical measurements of humidity , ambient temperature , pressure , wind direction , wind speed , viscosity etc. These pieces of information have been logged every specific amount of time say every 10 minutes or whatever. These massive amounts of data have been collected from the below data source:

Data source:

http://iot.ee.surrey.ac.uk/citypulse/datasets/weather/aarhus_weather_dewpoint

Data preparation

In order to have the data prepared to be processed by any frequent pattern algorithms it needs to be as follow:

- All fields are formatted in numeric format
- Every column has a unique numeric range

The encoding procedure will be as follow:

- Converting the timestamp column to numeric format
- keeping the humidity, dew point and pressure column
- adding 2000 to the temperature column
- adding 4000 to the wind direction column
- applying this formula for the wind speed=wind speed*10+6000

By applying this procedure to the raw data, the prepared data will be ready for processing by frequent pattern algorithms.

The IoT test results

By applying the three versions of code for the Apriori , FP-Growth , and the new method on the IoT dataset prepared data of 3,35,55,785 records , the output of the three versions of the programs are logged in Table 5.3 below. And the results are resumed in table 2 .It shows clearly that , the three versions of code representing the three methods come with the same results of the number of frequent count for all minimum supports (5%,1%,0.5%,0.25%,0.1%,0.05%). This proves the validity of the results of the new algorithm. Also,

as an average and overall , the new method satisfies better results in memory storage than the FP-Growth and better convergence time than the Apriori algorithms.

Furthermore, Fig. 2 depicts graphically the results of the three methods showing their time of execution as regard to the least support and proves the sample facts that the new method is better than the Apriori algorithm in time execution and better than FP-Growth as regard to the memory used in the execution.

Table 1: The output log of executing the IoT test case

Apriori	FP-Growth	The New Method
for support 5% APRIORI - Transactions count from database : 3,35,55,785 Frequent itemsets count : 15 Maximum memory usage : 535.17 MB Total time ~ 859 ms	for support 5% FP-GROWTH - Transactions count from database : 3,35,55,785 Frequent itemsets count : 15 Maximum memory usage 575.809 MB Total time ~ 782 ms	for support 5% New Algo.- Transactions count from database : 3,35,55,785 Frequent itemsets count : 15 Maximum memory usage 185.563 MB Total time ~: 796 ms
for support 1% Frequent itemsets count : 214 Maximum memory usage : 473.00 MB Total time ~ 468 ms	for support 1% Frequent itemsets count : 214 Maximum memory usage : 548.73 MB Total time ~ 1312 ms	for support 1% Frequent itemsets count : 214 Maximum memory usage : 219.7 MB Total time ~ 482 ms
for support 0.5% Frequent itemsets count : 519 Maximum memory usage : 473.0 MB Total time ~ 6422 ms	for support 0.5% Frequent itemsets count : 519 Maximum memory usage : 551.20 MB Total time ~ 1093 ms	for support 0.5% Frequent itemsets count : 519 Maximum memory usage : 311.20 MB Total time ~ 558ms
for support 0.25% Frequent itemsets count : 1432 Maximum memory usage : 428.7 MB Total time ~ 17965 ms	for support 0.25% Frequent itemsets count : 1432 Maximum memory usage : 385.67 MB Total time ~ 1078 ms	for support 0.25% Frequent itemsets count : 1432 Maximum memory usage : 299.1 9 MB Total time ~:594 ms
for support 0.1% Frequent itemsets count : 3418 Maximum memory usage : 428.7 MB Total time ~ 17816 ms	for support 0.1% Frequent itemsets count : 3418 Maximum memory usage 457.19 MB Total time ~ 1172 ms	for support 0.1% Frequent itemsets count : 3418 Maximum memory usage 390.76 MB Total time ~609 ms
for support 0.05% Frequent itemsets count : 8182 Maximum memory usage : 428.71 MB Total time ~ 28701 ms	for support 0.05% Frequent itemsets count : 8182 Maximum memory usage : 465.40 MB Total time ~ 1187 ms	for support 0.05% Frequent itemsets count: 8182 Maximum memory usage : 388.3 MB Total time ~ 698ms

Table 2: The results of Test case I in processing time

Transaction Count 121844 TIDs					
	Time of processing (ms)				
Support (%)	Apriori	FP-Growth	New (Proposed) algorithm	FP itemset count	Minimum support in transaction
5	859	782	796	15	6092
1	4688	1312	4822	214	1218

0.5	6422	1093	5585	519	610
0.25	17965	1078	5948	1432	305
0.1	17816	1172	6091	3418	122
0.05	28701	1187	6982	8182	61

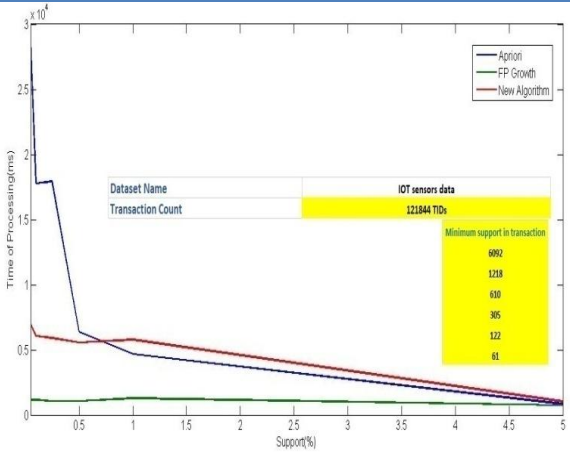

Fig 2: The output results of the test case I in processing time.

Table 3: The results of Test case I in Memory Usage

Transaction Count 121844 TIDs					
	Memory Usage (MB)				
Support (%)	Apriori	FP-Growth	New algorithm	FP itemset count	Minimum support in transaction
5	535	575	185	15	6092
1	473	548.7	219.7	214	1218
0.5	473	551.7	311.7	519	610
0.25	428.7	385	299	1432	305
0.1	428.7	457	390	3418	122
0.05	428.7	465	388	8182	61

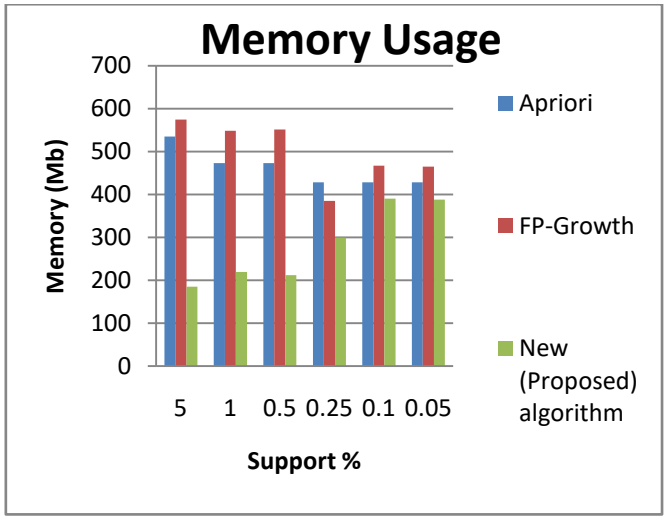

Fig 3: The output results of the test case I in Memory Usage

The above tables 2,3 and graph results figure 2,3 clearly showing that the new proposed algorithm is giving better performance than conventional algorithms in terms of time of processing and memory usage.

V. CONCLUSIONS AND FUTURE WORK

In this paper firstly, the heterogeneous big data has been identified and the problem has been named comprehensively. Consequently, all aspect of big data processing and frequent pattern mining have been analyzed immensely. The design for a new frequent pattern algorithm has been completed in two versions. One has been developed using C++ language and one by using MATLAB 2016. In addition to the development of the new algorithm there are two version have been implemented for two conventional method which are apriori and FP-growth for validation purposes to verify that the new algorithm are working accurately and gives the correct results.

The design of the proposed technique of frequent pattern mining, has considered two main aspects to be optimized which make the big difference for any good frequent pattern mining technique which are reducing the convergence time for speed up the execution and processing the big datasets and reducing the memory storage that has been used for the mining process of the big data.

Overall conclusion, the new method has a challenging performance as regard the memory storage and the speed of processing as compared to the conventional methods of frequent pattern mining like Apriori and FP-Growth techniques.

VI. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

REFERENCES

1. De Francisci Morales, G. (2013, May). SAMOA: A platform for mining big data streams. In Proceedings of the 22nd international conference on World Wide Web companion (pp. 777-778). International World Wide Web Conferences Steering Committee.,
2. Sun, Y., & Han, J. (2013). Mining heterogeneous information networks: a structural analysis approach. ACM SIGKDD Explorations Newsletter, 14(2), 20-28.
3. Paul, Sujni. "Parallel and distributed data mining." New Fundamental Technologies in Data Mining. InTech, 2011.
4. Da Silva, J. C., Giannella, C., Bhargava, R., Kargupta, H., & Klusch, M. (2005). Distributed data mining and agents. Engineering Applications of Artificial Intelligence, 18(7), 791-807.
5. Qian, M., & Zhai, C. (2014, November). Unsupervised Feature Selection for Multi-View Clustering on Text-Image Web News Data. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (pp. 1963-1966). ACM. Xiao Cai, Feiping And Nie and Heng Huang, "Multi View K-Means Clustering On Big Data Proceedings of the Twenty Third International Joint Conference On Artificial Intelligent
6. Motegaonkar, V. S., & Vaidya, M. V. (2014). A Survey on Sequential Pattern Mining Algorithms. International Journal of Computer Science and Information Technologies (IJCSIT), 5(2), 2486-2492.
7. R. Ahmed and G. Karypis, "Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks," Knowledge and Information Systems, vol. 33, no. 3, pp. 603-630, Dec. 2012.

8. M.H. Alam, J.W. Ha, and S.K. Lee, “Novel Approaches to Crawling Important Pages Early,” Knowledge and Information Systems, vol. 33, no. 3, pp 707-734, Dec. 2012.
9. S. Aral and D. Walker, “Identifying Influential and Susceptible Members of Social Networks,” Science, vol. 337, pp. 337-341, 2012.
10. A. Machanavajjhala and J.P. Reiter, “Big Privacy: Protecting
11. Confidentiality in Big Data,” ACM Crossroads, vol. 19, no. 1,pp. 20-23, 2012.
12. R. Agrawal, T. Imielinski, and A. Swami. Database Mining: A Performance Perspective. IEEE Transactions on Knowledge and Data Engineering, 5(6), pp. 914–925, 1993.
13. Nasreen, S., et al., (2014) " Frequent Pattern Mining Algorithms for Finding Associated Frequent Patterns for Data Streams: A Survey", Procedia Computer Science,37: p. 109-116.
14. Motegaonkar, V. S., & Vaidya, M. V. (2014). A Survey on Sequential Pattern Mining Algorithms. International Journal of Computer Science and Information Technologies (IJCSIT), 5(2), 2486-2492.
15. Fan, W., &Bifet, A. (2013). Mining big data: current status, and forecast to the future. ACM SIGKDD Explorations Newsletter, 14(2), 1-5.
16. Park, B. H., &Kargupta, H. (2002). Distributed data mining: Algorithms, systems, and applications.
17. Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining with big data. Knowledge and Data Engineering, IEEE Transactions on, 26(1), 97-107. (LR).
18. Gupta, R. (2014). Journey from Data Mining to Web Mining to Big Data. arXiv preprint arXiv:1404.4140.
19. Renjit, J. A., &Shunmuganathan, K. L. (2010). Mining the data from distributed database using an improved mining algorithm. arXiv preprint arXiv:1004.1677.
20. Tsai, C. W., Lai, C. F., Chiang, M. C., & Yang, L. T. (2014). Data mining for internet of things: A survey. Communications Surveys & Tutorials, IEEE, 16(1), 77-97.

AUTHORS PROFILE

Dr. Monika Saxena, Doctorate of Philosophy (Computer Science & Engineering), M.Tech (CSE) ,B.Tech (IT)
Assistant Professor, Banasthali Vidyapith, Bansthali
Rajsthan, India 304022

Prof. Chandra Kumar Jha , Doctorate of Philosophy (Computer Science)
Professor & Head, Computer Science Department, Banasthali Vidyapith, Bansthali
Rajsthan, India 304022.

Ms. Deepika Saxena, Research Scholar (Computer Science & Engineering), MCA
Computer Science Department, Sarvepalli Radhakrishnan University, Bhopal, India