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

Moore Data Clustering Based Bloom Hash Storage for Dimensionality Reduction of Big Data Analytics

Chitra. K, Maheswari. D

Abstract: ***Big data contains massive amounts of information's that are difficult to manage, acquire, store and analyses. The clustering of data is a demanding issue in the field of big data analytics. The existing techniques developed for clustering does not provide efficient performance and also time complexity of clustering was higher. Further, minimizing dimensionality of big data was not addressed effectively. In order to overcome these limitations, a Moore Data Clustering based Bloom Hash Storage (MDC-BHS) Technique is proposed. The MDC-BHS Technique is designed with aim of reducing the dimensionality of big data with lesser time through clustering. The MDC-BHS Technique used Moore Data Clustering (MDC) Model in order to group the data in big dataset with minimum time consumption. After performing clustering process, the MDC-BHS Technique employed Bloom Hash Storage (BHS) Model in order to store clustered data with minimum space complexity. The BHS Model is a space-efficient probabilistic data structure which utilized hashing function to create hash value for clustered data. Therefore, proposed MDC-BHS Technique significantly reduces the dimensionality of larger dataset. The experimental evaluation of MDC-BHS technique is carried out on weather data with factors such as clustering time and clustering accuracy and space complexity with respect to number of data. The experimental results demonstrate that MDC-BHS Technique is able to improve the clustering accuracy and also minimizes the space complexity when compared to state-of-the-art works.***

Keywords: ***Big Data, Bloom Hash Storage, Dimensionality Reduction, Hashing Function, Moore Clustering, Moore Curve***

I. INTRODUCTION

Clustering is an essential process in both machine learning and data mining. Clustering process groups the data of similar patterns to form the cluster. It is used in different applications such as pattern recognition, image analysis, information mining, bioinformatics and big data analytics. Big data contains huge volume of data. This increased size of data requires effective clustering techniques to reduce the memory use, and execution time.

Manuscript published on 30 September 2019
* Correspondence Author
Chitra. K*, Research Scholar, School of Computer Studies, Rathnavel Subramaniam College of Arts and Science, Sulur, Coimbatore, Tamil Nadu, Email: chitra.k@rvsgroup.com
Maheswari. D, Head, Research Coordinator, School of Computer Studies- PG, Rathnavel Subramaniam College of Arts and Science, Sulur, Coimbatore, Tamil Nadu, Email: maheswari@rvsgroup.com

In recent times, many research works are designed for clustering big data. But, the clustering performance of existing techniques was not sufficient.

In order to addresses the above mentioned existing issues, MDC-BHS Technique is developed. The main contributions of MDC-BHS Technique is formulated as follows,

To improve the clustering performance of big data with higher accuracy and minimum false positive rate, Moore Data Clustering (MDC) model is used in MDC-BHS Technique on the contrary to existing clustering concepts.

To reduce dimensionality of big data, Bloom Hash Storage (BHS) is employed in MDC-BHS Technique on the contrary to existing storage data structures. The BHS utilizes the hashing function for efficient big data storage. By using hashing function, BHS create the hash value for each clustered data in big dataset. Then, BHS stores hash value of data in it bit array instead of storing data. Therefore, BHS utilizes lesser amount of memory for effective big data analytics.

Therefore, this research work focuses on efficient clustering and dimensionality reduction of big data in order to achieve higher clustering accuracy and to reduce the space complexity

The rest of the paper is formulated as follows. The Section 2 explains the related works. In Section 3, the proposed MDC-BHS technique is explained with help of architecture diagram. The Experimental settings and comparative results analysis of proposed MDC-BHS technique is discussed in Section 4 and Section 5. Section 6 portrays the conclusion of paper.

II. RELATED WORKS

K-means Modified Inter and Intra Xlustering (KM-I2C) was presented in [1] to enhance clustering efficiency of big data with minimum execution times. The false positive rate of big data clustering was not solved in KM-I2C. Fuzzy consensus clustering (FCC) was developed in [2] for big data clustering. The dimensionality reduction of big data was not considered in FCC.

An incremental high order singular value decomposition (IHOSVD) method was designed in [3] for improving mining and analytics performance through minimizing dimensionality of big data. IHOSVD consumes more memory space for storing big data. A processing approach was intended in [4] based on feature extraction to minimize the dimensionality of microarray big data. A novel method was designed in [5] using Gaussian mixture model and principal component analysis for performing dimensionality reduction.

A comprehensive study of the partition based clustering algorithms developed for dimensionality reduction of big data was presented in [6]. Besides, review of different techniques designed using various data mining concepts for reducing dimensionality of big data was examined in [7]. A linguistic hedges neuro-fuzzy classifier was introduced in [8] to lessen dimensionality of medical big data. LHNFCSF was not solved the time complexity of dimensionality reduction. A high-order CFS algorithm was intended in [9] with aiming at grouping heterogeneous data with higher accuracy. The true positive rate of clustering using high-order CFS algorithm was poor.

A novel concept of big data reduction was designed in [10] to obtain multiple objectives. This concept not solves computational complexity of big data reduction. A fuzzy c-means (FCM) algorithm was introduced in [11] to enhance the clustering performance of very large data with higher clustering accuracy. The false positive rate of clustering using FCM algorithm was more. A De-duplication Aware Resemblance Detection and Elimination (DARE) Scheme were designed in [12] to attain big data reduction. DARE lacks efficiency of big data analytics. A Soft clustering was developed in [13] by combining fuzzy c-means and rough k-means in order to group the data in very large data sets with higher true positive rate. The soft clustering does not present optimal performance for big data clustering.

Fast Kernel Matrix Computation was presented in [14] improve the computation speed for clustering big data. The space complexity of Fast Kernel Matrix Computation was higher. The Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSIO-FCM) was introduced in [15] to address the challenges involved during big data clustering. The clustering performance of SRSIO-FCM was not efficient therefore lacks clustering accuracy. A two clustering validity indices was employed in [16] to group large amount of data with lower computational time. The big data reduction problem was remained unaddressed. A k-means algorithm was intended in [17] to cluster very large number of data with higher true positive rate. The clustering time of this method was very higher.

III. MOORE DATA CLUSTERING BASED BLOOM HASH STORAGE TECHNIQUE

Big data reduction is considered to be a significant problem as huge amount of big data introduces 'curse of dimensionality' with millions of features which improves the storage and computational complexity of big data. A wide range of dimension reduction methods are designed in the existing literature. The performances of existing clustering techniques were not effective. Besides, reducing dimensionality of big data was not solved efficiently. In order to solve these existing drawbacks, MDC-BHS Technique is introduced. The MDC-BHS Technique is designed with help of Moore clustering and bloom hash storage model for efficient big data analytics. The architecture diagram of MDC-BHS Technique is shown in below Figure 1.

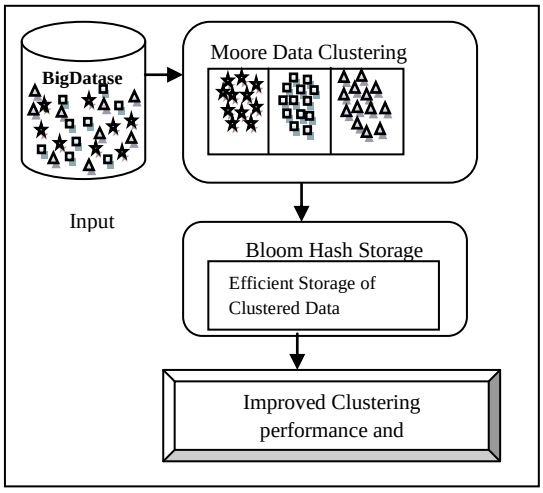

Fig.1. Architecture Diagram of MDC-BHS Technique for Dimensionality Reduction of Big Data

The MDC-BHS Technique at first takes big dataset as input. Moore Data Clustering (MDC) model is applied for grouping the large number of weather data into different clusters with higher accuracy and minimum time. Afterward, MDC-BHS Technique use Bloom Hash Storage (BHS) for storing the clustered big data with minimum space complexity.

A. Moore Data Clustering Model

The MDC-BHS Technique use Moore Data Clustering (MDC) Model with objective of grouping similar data in big dataset with minimum time. The MDC Model constructs the Moore curve to discover related data in dataset with high clustering accuracy. The Moore curve is a variation of the Hilbert curve. The differentiation in Moore curve is the locations of beginning and end of the curve. The Moore curve is a continuous path curve which passes through every block in a space once to form a one-one correspondence between the coordinates of blocks and the one-dimensional sequence numbers of blocks on curve.

The main goal of using MDC Model in MDC-BHS Technique is, it constructs Moore curve to preserve the distance of those data points which are close in space and also represents similar data close together in the linear order. This property of Moore curve helps MDC-BHS Technique to attain higher clustering. In addition to that, the mapping of data onto blocks in dimensional space minimizes the disk access effort and provides high speed for clustering big data. The Moore curve order 3 stores the similar data close together in the linear order. This assists for MDC Model to efficiently find out the similar types of data and to attain higher clustering performance for big data. The example structure of Moore curve order 3 is depicted in below Figure

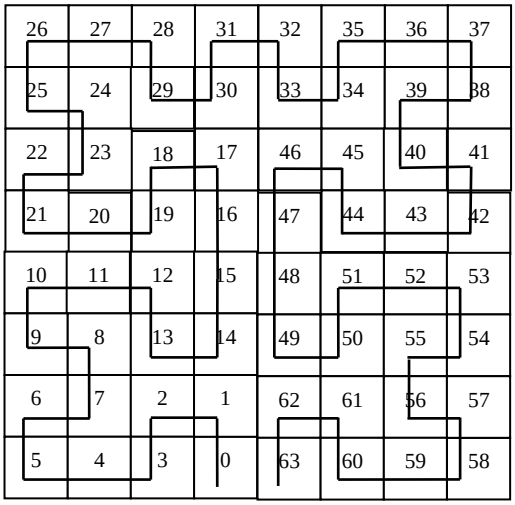

Fig. 2. Example Structure of Moore Curve Order 3.

Let us consider the El Nino Data Set that comprises of larger number of weather data that are represented as $DS = d_1, d_2, \dots, d_n$. The MDC Model at first maps the data objects in a given dataset onto blocks in dimensional space. The number of the data blocks $X[d_i]$ considered in MDC Model is denoted as n ($n = 2^{2M}$). Here, M represents the order of Moore curve. The size of block is based on the density of big data objects. Besides, $X[i] = 1$ denotes block i has data whereas $X[i] = 0$ indicates block i does not have data. The path of a Moore curve is a linear ordering that begins at one end of the curve and follows the path to the other end. The numbers in rectangle block of Moore curve refers the M-ordering. The MDC Model used Mapping function (φ) to map data points d_i onto dimensional space (DS) which expressed as,

$$\varphi: d_i \rightarrow DS \quad (1)$$

From equation (1), mapping of larger number of data onto a dimensional space is performed. As said by the property of Moore curve, if two blocks have data continuously, they are grouped in the same cluster. If not, they are clustered in various clusters. This process of MDC Model continual until all the data in rectangle blocks of Moore curve is clustered. The structure of Moore curve order 3 of big data clustering using El Nino Data Set is depicted in Figure 3.

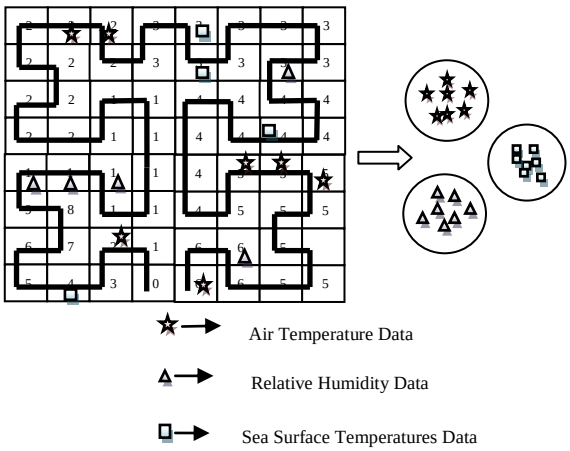

Fig. 3. Moore Curve Orders of 3 for Big Weather Data Clustering

Figure 3 presents the Moore Curve Orders of 3 to cluster the massive amount of weather information into different clusters. From figure, star, triangle and square denotes the clusters of air temperature data, relative humidity data, sea surface temperatures data in El Nino Data Set respectively. The algorithmic process of MDC Model for big data clustering is shown in below.

// Moore Data Clustering Algorithm

Input: El Nino Data Set ' $DS = d_1, d_2, \dots, d_n$' $X[i]$ is status of data block, cluster number $C_j = 1,2, \dots n$;

Output: Enhanced Clustering Accuracy and Reduced Time for big data analytics

Step 1:Begin

Step 2: Generate Moore Curve

Step 3: Maps all data in big dataset onto a rectangle blocks using (1)

Step 4: **For** each rectangle blocks

Step 5: **If** ($X[i]! = 0$) **then**

Step 6: **If** $X[i]$ and $X[i-1]$ have data continuously **then**

Step 7 $X[i] = C_j$; // group data in same cluster number

Step 8: **Else**

Step 9: $C_j = C_j + 1$; // group data in different cluster number

Step 10: **End if**

Step 11: **End if**

Step 12: **End For**

Step 13: End

Algorithm 1 Moore Data Clustering For Big Data Analytics

As demonstrated in algorithm 1, The MDC Algorithm initially generates Moore curve and then maps the all data in big dataset onto the rectangle blocks. If block i and the previous block contain data continuously, then data is clustered in same cluster number. Otherwise, the cluster number is incremented by one to cluster the data in different cluster. This is process repetitive until all the rectangle blocks of Moore curve is reached. This assists for MDC-BHS Technique to effectively cluster the different data in big dataset with higher accuracy. Hence, MDC-BHS Technique attains higher clustering accuracy and also minimum clustering time for analyzing big data.

B. Bloom Hash Storage based Big Data Dimensionality Reduction

After clustering, proposed MDC-BHS Technique use Bloom Hash Storage (BHS) for efficient and compact way of storing the big data. The BHS is a space-efficient probabilistic data structure which employs constant memory for storing clustered big data. On the contrary to existing storage data structures such as self-balancing binary search trees, hash tables, simple arrays and linked lists, the MDC-BHS Technique utilized BHS. Because BHS supports quick matching of membership query due to their randomized, space-efficient data structure with a reasonable or limited probability of occurrence of false answers. In addition to that, BHS takes constant time complexity for both inserting data and checking the membership of a data value in bit array.

The BHS used hash functions and a data bit set to identify the presence of data in bit array. Therefore, BHS consumes minimum amount of time to insert data and to check the occurrence of data as compared to storage structures used in conventional techniques. Furthermore, size of the bit set in BHS is much smaller when compared to the whole data. This helps for the MDC-BHS Technique to reduce the dimensionality of big data with minimum time. The process involved in BHS for minimizing the dimensionality of big data is shown in below.

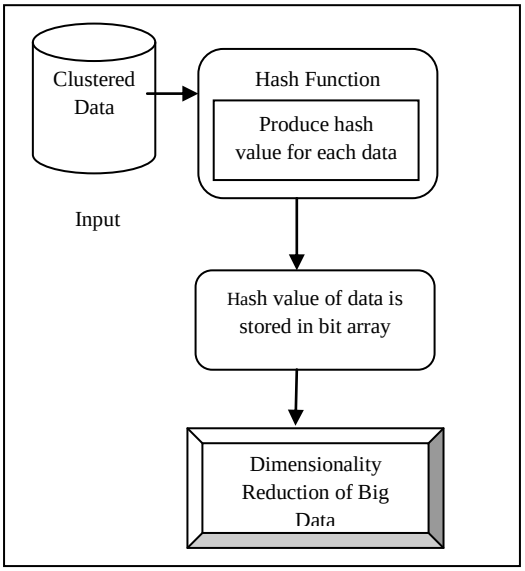

Fig. 4. Processes of Bloom Hash Storage for Dimensionality Reduction of Big Data

As shown in figure, BHS creates the hash value for each clustered data with aid of hash function. The BHS employed MurmurHash3 as hash function in MDC-BHS Technique. The MurmurHash3 generates 32-bit or 128-bit hash value for each data. Then, BHS stores generated hash value of data in it bit array. This supports for the MDC-BHS Technique to minimize space complexity of big data. Thus, MDC-BHS Technique achieves the dimensionality reduction for big data analytics.

Let us consider an empty BHS is a bit array 'BA' of 'l' bits and all set to 0. The MDC-BHS Technique store clustered big data $D_i = D_1, D_2, D_3, \ldots . D_n$ in 'BA' using hash functions. The hash function produces different hash value of the each clustered data to store it in bit array. The hash values of data is represented as ' $\omega = \omega_1, \omega_2, \omega_3, \ldots . \omega_n$. Let assume BHS selects each array position with equal probability to store the data. The BHS algorithm produces hash value for each data using below formulation,

$$HF = \omega_i(D_i) \quad (2)$$

From equation (2), HF denotes the hash function utilized in BHS where as ω_i represents the generated hash value of data D_i. Whenever a data is inserted in BHS, the probability that a certain bit is not set to 1 by a certain hash function represented as,

$$\left(1 - \frac{1}{l}\right) \quad (3)$$

From equation (3), 'l' refers the number of bits in 'BA'. Whenever 'α' is the number of hash functions, probability that the bit is not set to 1 by any of the hash functions expressed as,

$$\left(1 - \frac{1}{l}\right)^{\alpha} \quad (4)$$

After adding 'n' data, the probability that the bit is still zero formulated as,

$$\left(1 - \frac{1}{l}\right)^{\alpha n} \quad (5)$$

Correspondingly, probability of the bit is 1 obtained as,

$$1 - \left(1 - \frac{1}{l}\right)^{\alpha n} \quad (6)$$

In BHS, probability of false positive is measured using below expression,

$$\left(1 - \left(1 - \frac{1}{l}\right)^{\alpha n}\right)^{\alpha} \approx \left(1 - e^{-\frac{\alpha n}{l}}\right)^{\alpha} \quad (7)$$

The false positives probability of BHS is decreased when number of bits 'l' in the array increased or number of inserted data is increased. The proposed MDC-BHS technique minimizes the false positive probability of BHS by means of increasing number of inserted data as it considers big data for storage. Thus, the performance of BHS is optimized as,

$$BHS_{opt} = \frac{l}{n} \ln 2 \quad (8)$$

From equation (8), 'n' indicates the number of data stored in BHS. The MDC-BHS Technique increases the number of data stored in BHS by taking the big dataset. Hence, false positive probability of BHS is significantly reduced than a conventional bloom filter. As a result, BHS provides optimal performance for efficient and compact way of storing the big data and thereby reducing the dimensionality of big data. The following diagram shows structure of bloom hash storage.

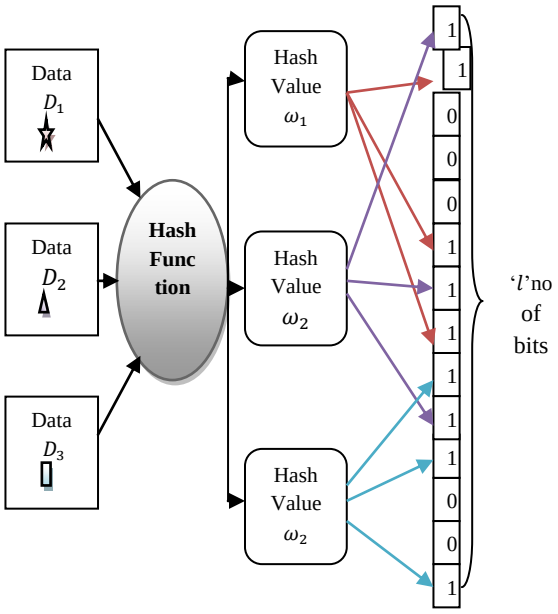

Fig. 5. Bloom Hash Storage Structure

Figure 5 depicts the bloom hash storage structure for dimensionality reduction of big data. When a BHS returns a false for query result (test whether data is in the BHS), it indicates that the data is not present in bit array. Hence BHS returns a true; it means that the data is presented in bit array. The algorithmic process of bloom hash storage is shown in below,

// Bloom Hash Storage Algorithm

Input: Cluster Data $D_i = D_1, D_2, D_3, \ldots . D_n$

Output: Reduced Space complexity for big data storage

Step 1: Begin

Step 2: For i to n do // each clustered data D_i

Step 3 Hash value of data ω_i is generated using (2)

Step 4: Store hash value of data in bit array

Step 5: End for

Step 6:End

Algorithm 2 Bloom Hash Storage for Dimensionality Reduction of Big Data

Algorithm 2 depicts step by step algorithmic process of BHS for efficient big data storage. The BHS stores hash values of clustered data in bit array. This hash value of data consumes smaller memory space and time for big data storage. This assists for BHS to reduce the space complexity of big data with minimum time utilization. As a result, proposed MDC-BHS Technique minimizes the dimensionality of big data.

IV. EXPERIMENTAL SETTINGS

In order to measure the performance, MDC-BHS Technique is implemented in Java language using El Nino Data Set. The MDC-BHS Technique used El Nino Data Set [18] obtained from UCI machine learning repository for conducting experimental evaluation. This El Nino Data Set includes of oceanographic and surface meteorological data. The El Nino Data Set consists of 178080 numbers of instances and 12 attributes. The following are attributes of El Nino Data Set such as date, latitude, longitude, zonal winds, meridional winds, relative humidity, air temperature, sea surface temperature and subsurface temperatures. The MDC-BHS Technique effectively clusters the various weather data with higher accuracy and minimum false positive rate and time using MDC model. Besides, MDC-BHS Technique also lessens the dimensionality of El Nino Data Set for performing big analytics with lower space complexity with aid of bloom hash storage.

The experimental evaluation of MDC-BHS Technique is carried out for many instances with respect to different number of weather data and averagely ten results are exposed in table and graph for analyzing proposed performance. The effectiveness of MDC-BHS Technique is measured in terms of clustering accuracy, space complexity, clustering time and false positive rate.

V. RESULTS AND DISCUSSIONS

The comparative result of MDC-BHS Technique is presented in this section. The efficacy of MDC-BHS Technique is compared against with existing K-means modified inter and intra clustering (KM-I2C) [1] and Fuzzy Consensus Clustering (FCC) [2] respectively. The performance results of MDC-BHS Technique are analyzed with helps of the following metrics.

A. Performance Result of Clustering Accuracy

In MDC-BHS technique, Clustering accuracy CA is measured as ratio of number of data that are correctly clustered to the total number of data taken as input. The clustering accuracy is determined in terms of percentage (%) and mathematically formulated as,

$$CA = \frac{Number\ of\ data\ correctly\ clustered}{total\ number\ of\ data\ taken\ as\ input} * 100 \qquad (9)$$

From equation (9), clustering accuracy of big data is evaluated. When clustering accuracy is higher, the MDC-BHS technique is said to be more effectual.

To measure the clustering accuracy, the MDC-BHS considers the technique with different number of data in the range of 100-1000 for conducting experimental process. When considering 600 number of weather data for experimental evaluation, proposed MDC-BHS technique achieves 90% clustering accuracy whereas existing KM-I2C [1] and FCC [2] obtains 71% and 82% respectively. From that, it is illustrative that the clustering accuracy using proposed MDC-BHS technique is higher. The below Table 1 illustrates the performance result analysis of clustering accuracy for handling large number of data

Table 1 Tabulation Result of Clustering Accuracy

Number of Data	Clustering Accuracy (%)		
	KM-I2C	FCC	MDC-BHS
100	61	69	82
200	65	74	85
300	66	77	86
400	68	78	87
500	69	79	89
600	71	82	90
700	72	83	91
800	75	84	93
900	76	85	94
1000	78	87	95

Table 1 shows the impacts of clustering accuracy with respect to various numbers of data in the range of 100-1000 using three methods namely KM-I2C [1], FCC [2] and proposed MDC-BHS technique. The proposed MDC-BHS technique provides better clustering accuracy for grouping large number of data in big dataset as compared to existing works KM-I2C [1], FCC [2]. This is because of application of MDC model in MDC-BHS technique where it formulates Moore curve for effectively clustering the numerous data in big El Nino Data Set. Therefore, proposed MDC-BHS technique increases the clustering accuracy of big data by 28% and 12% when compared to KM-I2C [1], FCC [2] respectively.

B. Performance Result of Space Complexity

In MDC-BHS technique, Space complexity SC determines amount of memory space needed to store the clustered data. The space complexity is evaluated in terms of Mega bytes (MB) and measured as follows,

$$SC = n * memory\ (storing\ data) \qquad (10)$$

From equation (10), space complexity of big data is measured. Here, n denotes number of data taken as input. When space complexity is lower, the technique is said to be more efficient.

MDC-BHS technique acquires 99 MB space complexity whereas existing KM-I2C [1] and FCC [2] gets 142 MB and 120 MB respectively. From these results, it is expressive that the space complexity using proposed MDC-BHS technique is lower for efficient big data analytics. The below Figure 6 depicts the experimental results of space complexity using three methods for analyzing big data.

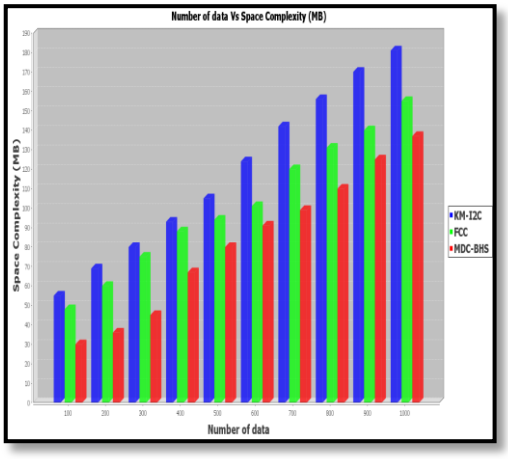

Fig. 6. Comparative Result of Space Complexity

Figure 6 portrays the comparative result analysis of space complexity for big data based on diverse numbers of data in the range of 100-1000 using three methods namely KM-I2C [1], FCC [2] and proposed MDC-BHS technique. The proposed MDC-BHS technique provides better space complexity as compared to existing works KM-I2C [1], FCC [2]. This is owing to application of BHS in proposed MDC-BHS technique where it uses hash value of data for efficient big data storage. As a result, proposed MDC-BHS technique reduces the space complexity of big data by 33% and 22% when compared to KM-I2C [1], FCC [2] respectively.

C. Performance Result of Clustering Time

In MDC-BHS technique, Clustering Time CT measures amount of time required for grouping the big data. The clustering time is estimated in terms of milliseconds (ms) and represented as below,

$$CT = n * time\ (clustering\ data) \qquad (11)$$

From equation (11), the amount of time needed for clustering big data is determined. Here, n indicates the number of data considered as input. When clustering time is lower, the technique is said to be more effective.

For determining the amount of time taken for clustering the big data, MDC-BHS takes the different number of data in the range of 100-1000 for experimental evaluation. When assuming 800 number of weather data from El Nino Data Set for accomplishing experimental process, proposed MDC-BHS technique takes 53ms clustering time whereas existing KM-I2C [1] and FCC [2] gets 68ms and 64ms respectively. According, it is clear that the clustering time of big data using proposed MDC-BHS technique is lower compared to other existing methods. The following shows presents the tabulation results of clustering time using three methods for handling large size of El Nino Data Set dataset.

Table 2 Tabulation Result of Clustering Time

Number of Data	Clustering Time (ms)		
	KM-I2C	FCC	MDC-BHS
100	40	37	24
200	44	42	27
300	49	45	31
400	52	50	36
500	58	53	39
600	62	57	45
700	66	61	49
800	68	64	53
900	73	69	57
1000	79	75	61

Table 2 describes the experimental result analysis of time required for clustering the big weather data versus various numbers of data in the range of 100-1000 using three methods namely KM-I2C [1], FCC [2] and proposed MDC-BHS technique. The proposed MDC-BHS technique provides better clustering time for big weather data analytics when compared to existing works KM-I2C [1], FCC [2]. Hence, proposed MDC-BHS technique reduces the clustering time of big data by 32 % and 22 % when compared to KM-I2C [1], FCC [2] respectively.

D. Performance Result of False Positive Rate

In MDC-BHS technique, False Positive Rate FPR determined as ratio of number of data that are incorrectly clustered to the total number of data taken as input. The false positive rate of clustering is measured in terms of percentage (%) and mathematically obtained as,

$$FPR = \frac{Number\ of\ data\ correctly\ clustered}{total\ number\ of\ data\ taken\ as\ input} * 100 \qquad (12)$$

From equation (12), false positive rate of big data clustering is determined. When false positive rate of clustering is lower, the technique is said to be more effective.

Let us consider MDC-BHS assumes the dissimilar number of weather data in the range of 100-1000 to measure false positive rate of big data. When taking 900 number of weather data from El Nino Data Set for experimental work, proposed MDC-BHS technique attains 49% false positive rate whereas existing KM-I2C [1] and FCC [2] obtains 70% and 62% respectively. Thus, it is descriptive that the false positive rate of big data clustering using proposed MDC-BHS technique is lower as compared to other existing methods. The below figure illustrates graph representation of performance results of false positive rate for clustering large size of dataset using three methods.

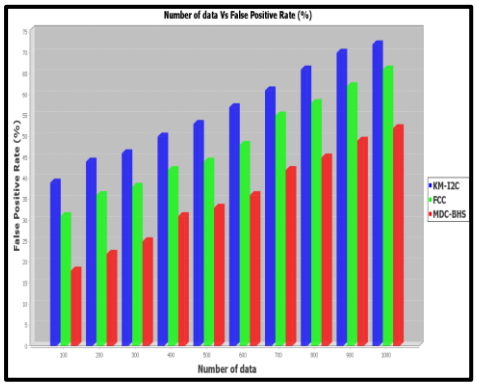

Fig. 7. Comparative Result of False Positive Rate

Figure 7 explains the performance result analysis of false positive rate of big data clustering along with different numbers of data in the range of 100-1000 using three methods namely KM-I2C [1], FCC [2]

and proposed MDC-BHS technique. Moreover, the false positive rate of big data clustered is enhanced with increasing the number of input weather data using all the three techniques. As a result, proposed MDC-BHS technique minimizes the false positive rate of big data by 38 % and 28 % when compared to KM-I2C [1], FCC [2] respectively.

As a result, MDC-BHS Technique is designed in order to attain higher clustering and dimensionality reduction performance for big data analytics.

VI. CONCLUSION

The proposed MDC-BHS Technique is developed with objective of enhancing clustering and dimensionality reduction performances of big data with lesser time. The objective of MDC-BHS Technique is achieved with MDC Model and BHS. The MDC Model supports for MDC-BHS Technique to group the data in big dataset with lower false positive rate through constructing Moore curve. From that, MDC-BHS Technique efficiently clusters the various types of data in big dataset into different clusters with higher accuracy and minimum time. Therefore, MDC-BHS Technique attains higher clustering accuracy and reduced false positive rate and clustering time for big data analytics as compared to state-of-art-works. Furthermore, BHS assists for MDC-BHS Technique to store clustered big data with minimum space complexity. As a result, proposed MDC-BHS Technique attains dimensionality reduction for efficient big data analytics when compared to state-of-art-works. The performance of MDC-BHS Technique is tested with the metrics such as clustering accuracy, space complexity, clustering time and false positive rate. With the experiments conducted for MDC-BHS Technique, it is clear that the cluster accuracy provides more accurate results and provides better performance with a reduction of space complexity and the reduction of false positive rate of big data clustering when compared to state-of-the-art works.

REFERENCES

1. Chowdam Sreedhar, Nagulapally Kasiviswanath, Pakanti Chenna Reddy, “Clustering large datasets using K-means modified inter and intra clustering (KM-I2C) in Hadoop”, Journal of Big Data, Springer, Volume 4, Issue 27, pp. 1-19, 2017
2. Junjie Wu, Zhiang Wu, Jie Cao, Hongfu Liu, Guoqing Chen, Yanchun Zhang, “Fuzzy Consensus Clustering With Applications on Big Data”, IEEE Transactions On Fuzzy Systems, Volume 25, Issue 6, Pp. 1430 – 1445, December 2017
3. Liwei Kuang, Fei Hao, Laurence T. Yang, Man Lin, Changqing Luo, Geyong Min, “A Tensor-Based Approach for Big Data Representation and Dimensionality Reduction” IEEE Transactions on Emerging Topics in Computing, Volume 2, Issue 3, Pp. 280 – 291, 2014
4. Fadoua Badaoui, Amine Amar, Laila Ait Hassou, Abdelhak Zoglat, Cyrille Guei Okou, “Dimensionality reduction and class prediction algorithm with application to microarray Big Data”, Journal of Big Data, Springer, Volume 4, Issue 32, Pp. 1-11, December 2017
5. Ewa Nowakowska, Jacek Koronacki, Stan Lipovetsky, “Dimensionality reduction for data of unknown cluster structure”, Information Sciences, Elsevier, Volume 330, Pp. 74-87, February 2016
6. Vallabh Dhoot, Shubham Gawande, Pooja Kanawade ,Akanksha Lekhwani, “Efficient Dimensionality Reduction for Big Data Using Clustering Technique”, Imperial Journal of Interdisciplinary Research, Volume 2, Issue 5, Pp. 26-29, 2016
7. Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, Samee U. Khan, “Big Data Reduction Methods: A Survey, Data Science and Engineering, Springer, Volume 1, Issue 4, Pp. 265–284, December 2016
8. Ahmad Taher Azar, Aboul Ella Hassanien, “Dimensionality reduction of medical big data using neural-fuzzy classifier”, Soft Computing, Springer, Volume 19, Issue 4, Pp. 1115–1127, April 2015
9. Fanyu Bu, Zhikui Chen, Peng Li, Tong Tang, and Ying Zhang, “A High-Order CFS Algorithm for Clustering Big Data”, Mobile Information Systems, Hindawi, Volume 2016, Article ID 4356127, Pp. 1-8, 2016
10. Muhammad Habib ur Rehmana, Victor Chang, Aisha Batool, Teh Ying Wah, “Big data reduction framework for value creation in sustainable enterprises”, International Journal of Information Management, Elsevier, Volume 36, Pp. 917–928, 2016
11. Timothy C. Havens, James C. Bezdek, Christopher Leckie, Lawrence O. Hall, Marimuthu Palaniswami, “Fuzzy c-Means Algorithms for Very Large Data”, IEEE Transactions on Fuzzy Systems, Volume 20, Issue 6, Pp. 1130 – 1146, 2012
12. Wen Xia, Hong Jiang, Dan Feng, Lei Tian, “DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads”, IEEE Transactions on Computers, Volume 65, Issue 6, Pp. 1692 – 1705, 2016
13. Min Chen, “Soft Clustering for Very Large Data Sets”, IJCSNS International Journal of Computer Science and Network Security, Volume 17, Issue 1, Pp. 102-108, January 2017
14. Nikolaos Tsapanos, Anastasios Tefas, Nikolaos Nikolaidis, Alexandros Iosifidis, and Ioannis Pitas, “Fast Kernel Matrix Computation for Big Data Clustering”, Procedia Computer Science, Elsevier, Volume 51, Pp. 2445–2452, 2015
15. Neha Bharill, Aruna Tiwari, Aayushi Malviya, “Fuzzy Based Scalable Clustering Algorithms for Handling Big Data Using Apache Spark”, IEEE Transactions on Big Data, Volume 2, Issue 4, Pp. 339 – 352, 2016
16. Jose Maria Luna-Romera, Jorge García-Gutiérrez, Maria Martínez-Ballesteros, Jose C. Riquelme Santos, “An approach to validity indices for clustering techniques in Big Data”, Progress in Artificial Intelligence, Springer, Pp. 1–14, 2017
17. Mugdha Jain, Chakradhar Verma, “Adapting k-means for Clustering in Big Data”, International Journal of Computer Applications, Volume 101, Issue 1, Pp. 19-24, September 2014
18. El Nino Data Set: https://archive.ics.uci.edu/ml/datasets/El+Nino

AUTHORS PROFILE

K. Chitra received her B.Sc Computer Technology from Coimbatore Institute of Technology, Coimbatore, India. She had her M.Sc Computer Communication from Bharathiar University, Coimbatore, India. She holds M.Phil in Computer Science from Bharathiar University, Coimbatore, India. She has 9 years of experience in teaching. She is presently working as an Assistant Professor in Rathnavel Subramaniam College of Arts and Science, Coimbatore. Her research interest includes Data Structures, Data Mining and Big Data Analytics. Now she is pursuing her Ph.D Computer Science in Rathnavel Subramaniam College of Arts and Science, Coimbatore.

Dr. D. Maheswari received her M.Sc Computer Science from Avinashilingam University for Women, Coimbatore, India. She completed her M.Phil and Ph.D degree in Computer Science in Avinashilingam University for Women, Coimbatore, India. She has been working as Head & Research Coordinator in School of Computer Studies in Rathnavel Subramaniam College of Arts and Science, Coimbatore, India. She has published more than 40 papers in International / National Journal and Conferences. Her research work focuses on Image processing and Data Mining. She has 9 years of teaching experience and 9 years of research experience.