International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 8813-8823

Generalized Jaccard Similarity Based Multilevel Threshold Affinity Propagated Clustering For Big Data Analytics

Maheswari K[*], Dr.Ramakrishnan M[b]

[a]*Research Scholar, Department of Computer Science, Bharathiyar University, Coimbatore - 641046, Tamil Nadu, India.*
[b]*Chairperson, School of Information Technology, Madurai Kamaraj University, Madurai, Tamil Nadu, India.*

Abstract

Clustering the huge amount of information in very large datasets is a difficult problem to be solved in data mining. Few research works have been designed for grouping similar types of data in big dataset with aid of different data mining concepts. The computational cost of conventional affinity propagation clustering technique is expensive in terms of memory space and time complexities when considering large size of dataset as input. In order to overcome these limitations, Generalized Jaccard Similarity based Multilevel Threshold Affinity Propagated Clustering (GJS-MTAPC) Technique is proposed. The GJS-MTAPC Technique is an improved Affinity propagation (AP) algorithm to increase the clustering performance of big data with minimal false positive rate and minimal computational cost. The GJS-MTAPC Technique splits big dataset which is to be clustered into a number of subsets. After dividing dataset, GJS-MTAPC Technique chooses exemplars of each subset randomly. Then, GJS-MTAPC Technique identifies best exemplars by transmitting responsibility and availability messages among data samples in each subset. Finally, GJS-MTAPC Technique defines multiple threshold values in order to precisely cluster data samples in a large dataset based on similarity values. As a result, GJS-MTAPC Technique provides better big data clustering processes in terms of clustering accuracy, computational cost and space complexity and false positive rate. The experimental result show that GJS-MTAPC Technique is able to increases the clustering accuracy and also minimizes the computational cost of big data analytics as compared to state-of-the-art works.

Keywords: Big data, Exemplars, Clusters, Generalized Jaccard Similarity, Multiple Threshold Values, Subsets

INTRODUCTION

A huge size of data is collected everyday due to the rising involvement of humans in the digital space. Such huge amount of data includes useful information is termed as Big Data. Big data analytics is popularly increase to acquire valuable information that can be of great use in scientific and business applications. With the rapid speed of internet development, people get more focus on the Big Data issue. Big Data is difficult to analyze. One of the common methods that help to analyzing data is cluster analysis. Clustering is the process of grouping data where the members of group are similar. Efficient clustering is the demanding problem in data mining techniques because the availability of large size of dataset. The existing clustering algorithm takes more clustering time and also requires huge amount of memory space to group big data. Hence, this research work focuses on improving the clustering accuracy and reduces the clustering time as well as computation cost of big dataset.

Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSIO-FCM) was presented in [1] in order to solve the problems interrelated with big data clustering. The SRSIO-FCM lessens the amount of time required for big data clustering. The quality of clustering using SRSIO-FCM was poor. A Weighted consensus fuzzy clustering (WCFC) was introduced in [2] with intention of increasing speed of large-scale data clustering. The false positive rate of WCFC was not solved.

A clusiVAT algorithm was designed in [3] with objective of clustering big data with higher accuracy. The time complexity of big data clustering was very higher. K‑means modified inter and intra clustering (KM-I2C) was developed in [4] to perform big data clustering with lower execution time. The clustering performance of KM-I2C was not effectual.

A MapReduce-based artificial bee colony (MR-ABC) was intended in [5] to attain large-scale data clustering. The clustering accuracy of MR-ABC was lower. A k-means algorithm was presented in [6] for clustering big data and to attain higher true positive rate. The time utilized to grouping the data was more.

A Fuzzy consensus clustering (FCC) was intended in [7] to carry out big data clustering. FCC requires more execution time. A two clustering validity indices was presented in [8] to cluster huge volume of data with minimal computational time. The space complexity of big data clustering was not solved.

A Soft clustering algorithm was presented in [9] through integrating fuzzy c-means and rough k-means to attain higher clustering performance of big data. The computational complexity of soft clustering algorithm was remained an open issue. Fast Kernel Matrix Computation was designed in [10] in order to minimize computational speed for grouping big data. This Fast Kernel Matrix Computation does not present better clustering efficiency.

In order to overcome above said existing drawbacks of conventional big data clustering, GJS-MTAPC Technique is introduced. The contributions of GJS-MTAPC Technique is organized as follows,

- To attain higher clustering performance for big data analytics as compared to state-of-the-art works, GJS-MTAPC Technique is designed with applications of Generalized Jaccard Similarity Coefficient measurement and Multilevel Threshold values.

❖ To reduce time complexity and space complexity of existing affinity propagation clustering, Generalized Jaccard Similarity Coefficient and Multilevel Threshold values are used in GJS-MTAPC technique. The Generalized Jaccard Similarity Coefficient helps for GJS-MTAPC Technique to measure the relatedness between data samples to significantly cluster the big data. The Multilevel Threshold values assists for GJS-MTAPC Technique to effectively form number of clusters during big clustering process.

The rest of the paper is planned as follows. Section 2 describes the existing clustering techniques designed for big data. In Section 3, the proposed GJS-MTAPC Technique is explained with assist of neat architecture diagram. The experimental settings and performance results analysis of proposed technique is presented in Section 4 and Section 5. Section 6 reveals the conclusion of paper.

RELATED WORKS

A lot of research works designed for big data clustering with help of data mining concepts. A high-order CFS algorithm was designed in [11] in order to cluster heterogeneous data with higher precision. The false positive rate of clustering was not addressed in high-order CFS algorithm. A fuzzy c-means (FCM) algorithm was employed in [12] for clustering large data and increasing clustering accuracy. The FCM algorithm needs more processing time.

K-Means and K-Medoids algorithm was presented in [13] based on number of clusters constructed using distance metric for analyzing big data. The time complexity involved during big data clustering was more. Fast Constrained Spectral clustering technique was introduced in [14] with goal of enhancing the clustering efficiency of big data with lower space and time complexity.

A Sparse Self-Represented Network Map was intended in [15] for clustering the data in large dataset. The clustering accuracy of large data was not at required level. Clustering-based Collaborative Filtering approach (ClubCF) was used in [16] for grouping similar data in big dataset. The precision and time complexity of ClubCF approach was not adequate.

Sketch and-validate (SkeVa) framework was introduced in [17] for clustering the large dataset into dissimilar cluster with minimal time. The performance result of true positive rate was not considered in SkeVa. The Parallel Clustering Algorithm was designed in [18] to increases the clustering accuracy of large biological data.

The review of iterative big data clustering algorithms was presented in [19]. A scalable machine-learning algorithm was explained in [20] for addresses problem involved during big data analytics. The time and space complexity of big data was not solved efficiently.

To solve above mentioned existing issues of conventional big data clustering, GJS-MTAPC Technique is proposed which is explained below section.

GENERALIZED JACCARD SIMILARITY BASED MULTILEVEL THRESHOLD AFFINITY PROPAGATED CLUSTERING TECHNIQUE

Clustering is a significant data mining technique which is widely used for mining valuable information. The objective of clustering is to separate the data into groups in which data in each group are similar to each other and differ from data in other groups. Over the past decades, a lot of clustering algorithms are designed based on different data mining techniques. AP is a one of the efficient clustering method. The AP clustering begins with computing similarity between data and then a message is transmitted to all data samples to select best exemplars and clusters formation. The conventional AP clustering algorithm attains higher efficiency and accuracy for clustering data. However, AP clustering is not suitable for large size of dataset as it takes more memory and time. To attain both a low computational cost and a good accuracy during big data clustering processes, Generalized Jaccard Similarity based Multilevel Threshold Affinity Propagated Clustering (GJS-MTAPC) Technique is introduced.

On the contrary to different existing clustering algorithm in data mining, proposed GJS-MTAPC Technique considers the AP clustering because it does not initialize the number of clusters. Furthermore, AP algorithm includes various advantages compared with existing clustering methods such as speed, good clustering performance and no need for clustering number parameter. On the contrary to k-means algorithm, AP assumes all data samples as exemplars concurrently and thus discovers best exemplars to cluster the data with higher accuracy. Therefore, an improved AP algorithm (i.e. GJS-MTAPC Technique) is proposed.

The GJS-MTAPC Technique is designed with a help of Generalized Jaccard Similarity Coefficient measurement and Multilevel Threshold values when compared to existing clustering techniques. The Generalized Jaccard Similarity Coefficient is a statistic used for evaluating the similarity between data samples. By using Generalized Jaccard Similarity Coefficient measurement, GJS-MTAPC Technique measure similarity between pair of data samples which indicates how the data samples in big dataset are related. The high values point out higher similarity between data samples whereas low values refer lower similarity between data samples in big dataset. With aid of measured similarity values between data samples and exemplars, GJS-MTAPC Technique finds best exemplars for whole big dataset for efficiently forming number of clusters with higher clustering accuracy.

Then, GJS-MTAPC Technique initializes multiple threshold values with helps of similarity values of identified best exemplars to increases the big data clustering performances as compared to existing clustering techniques. The existing AP algorithm used distance between data samples as similarity to perform clustering processes whereas GJS-MTAPC Technique considers relatedness between data samples as similarity. This helps for GJS-MTAPC Technique to group the symmetric types of data in very large dataset based on their similarity value with higher accuracy and minimal amount of time utilization. Therefore, GJS-MTAPC

Technique lessens the false positive rate and computation cost of big data clustering in an effective manner. The architecture diagram of GJS-MTAPC Technique for clustering data in big dataset is demonstrated in below Figure 1.

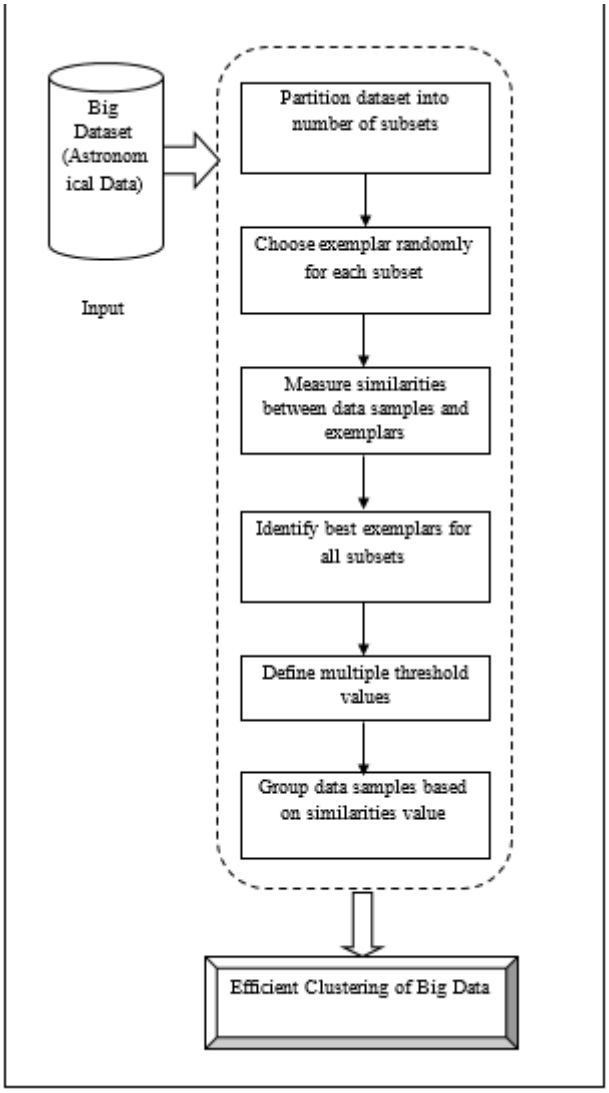

Figure 1. Architecture Diagram of Generalized Jaccard Similarity based Multilevel Threshold Affinity Propagated Clustering Technique for Big Data

Figure 1 illustrates flow processes of GJS-MTAPC Technique for enhancing clustering performance of big data. As presented in Figure 1, GJS-MTAPC Technique at first takes big dataset (i.e. astronomical data) as input. Then, the big dataset is splits into number of subsets. Afterward, GJS-MTAPC Technique selects exemplars of each subset randomly to find out best exemplars for efficient big data clustering processes. The GJS-MTAPC Technique selects best exemplar in each subset by iteratively sending two kinds of messages such as responsibility and availability among data samples. By means of passing the messages to all data samples in subset, GJS-MTAPC Technique finds best exemplars of whole data set. Subsequently, GJS-MTAPC Technique sets multiple threshold values to efficiently and accurately cluster data samples in a large dataset. Finally, GJS-MTAPC Technique groups all the data samples according to similarity values between each data samples and exemplars. Hence, GJS-MTAPC Technique gets higher clustering accuracy and minimal computational cost for big data clustering. The elaborate process of GJS-MTAPC Technique is described in below.

Let us consider a big dataset 'DS' that comprises huge number of different types of data sample represented as '$DS = D_1, D_2, D_3, \dots D_N$'. Here, N denotes the total number of data samples in big dataset. The proposed technique groups the data sample in big dataset into a different clusters using GJS-MTAPC. The GJS-MTAPC technique at first splits the big dataset to be clustered into a number of subsets which is obtained as,

$$DS = \{ SS_1, SS_2, \dots SS_n \} \quad (1)$$

From equation (1), '$SS_1, SS_2, \dots SS_n$' represent the number of subsets like $SS_1 = (D_1, D_2, D_{3,..})$, $SS_2 = (D_4, D_5, D_{6,..})$, …. $SS_n = (D_7, D_9, .. D_N)$ whereas 'D_i' refers the data sample in big dataset. After partitioning dataset, GJS-MTAPC Technique randomly elects exemplar 'x' for each subset 'SS_i'which is formulated as,

$$x = Random\ (D_i) \in SS_i \quad (2)$$

From equation (2), exemplar of each subset is selected randomly. The similarity between data sample D_i and exemplars x is determined with help of generalized Jaccard coefficient as below

$$S(D_i, x) = \frac{\sum_{i=1}^{n} \min(D_i, x)}{\sum_{i=1}^{n} \max(D_i, x)} \quad (3)$$

From equation (3), similarity between data samples and exemplar '$S(D_i, x)$' is estimated for each subset in big dataset. Here, 'n' denotes number of data samples in subsets. The data sample with larger similarity value in subset is selected as best exemplar for each subset 'SS_i' by means of sending messages between data samples. The GJS-MTAPC technique employed two types of messages namely availability message, and responsibility message to find best exemplar of subsets. The GJS-MTAPC technique updates these two messages iteratively to discover best exemplar and to form the number of clusters during big data clustering process.

The responsibility message transmitted from data sample 'D_i' to exemplar point 'x' which denotes how suitable the data sample 'x' be identified as a best exemplar within a subset 'SS_i'. The updating rule of responsibility message is expressed as

$$r(D_i, x) \leftarrow s(D_i, x) - \max_{x' \, s.t \, x' \neq x} \{ a(D_i, x') + s(D_i, x') \} \quad (4)$$

From equation (4), '$r(D_i, x)$' refers the responsibility message. Here, $S(D_i, x)$ denotes similarity matrix between data points and exemplar. Besides, the availability message broadcasted from exemplar point 'x' to data sample 'D_i' signifies how accurately for point 'D_i' find outs 'x' as the best exemplar. The updating rule of availability message is represented as,

$$a(D_i, x) \leftarrow \min\{0, r\ (x, x) +$$

$$\sum_{D_i' \, s.t \, D_i' \notin \{D_i, x\}} \max\{0, r(D_i', x)\}\} \quad (5)$$

From equation (5), '$a(D_i, x)$' refers the responsibility message. The responsibilities and availabilities are updated recurrently until a termination condition is reached. The process of finding best exemplar is terminated in GJS-MTAPC technique after a maximum number of iterations. After the iteration, each subset identifies its best exemplar. For each data sample 'D_i' in subset, the data sample which has higher similarity value among data samples and exemplar is selected as best exemplar 'X' which formulated as,

$$X = \arg\max \, S(D_i, x) \quad (6)$$

From equation (6), best exemplar within a subset is identified to group the similar kinds of data samples in big dataset with higher accuracy. With aid of similarity value of identified best exemplar, the GJS-MTAPC technique defines multiple similarity threshold values for efficient formation of clusters which is expressed as,

$$\delta_i - \delta_j \Rightarrow C_1 \quad (7)$$

$$\delta_k - \delta_l \Rightarrow C_2 \quad (8)$$

$$\delta_m - \delta_n \Rightarrow C_3 \quad (9)$$

From above equations (7), (8), (9), multiple similarity threshold values $\delta_i - \delta_j$,$\delta_k - \delta_l$, $\delta_m - \delta_n$ are defined in order to efficient clustering of data in big dataset with higher accuracy and minimal time consumption. Thus clustering result of big data is formulated as,

$$DS = C_1 \cup C_2 \cup C_3 \ldots \cup C_n \quad (10)$$

From equation (10), grouping results of similar types of data in very large dataset is obtained. The GJS-MTAPC Technique considers the big astronomical data for carried outing clustering process which comprises huge volume of information about pulsar candidates gathered during HTRU survey. For a big astronomical data, the GJS-MTAPC Technique obtains two clusters namely pulsar data and non-pulsar data which are depicted in below Figure 2.

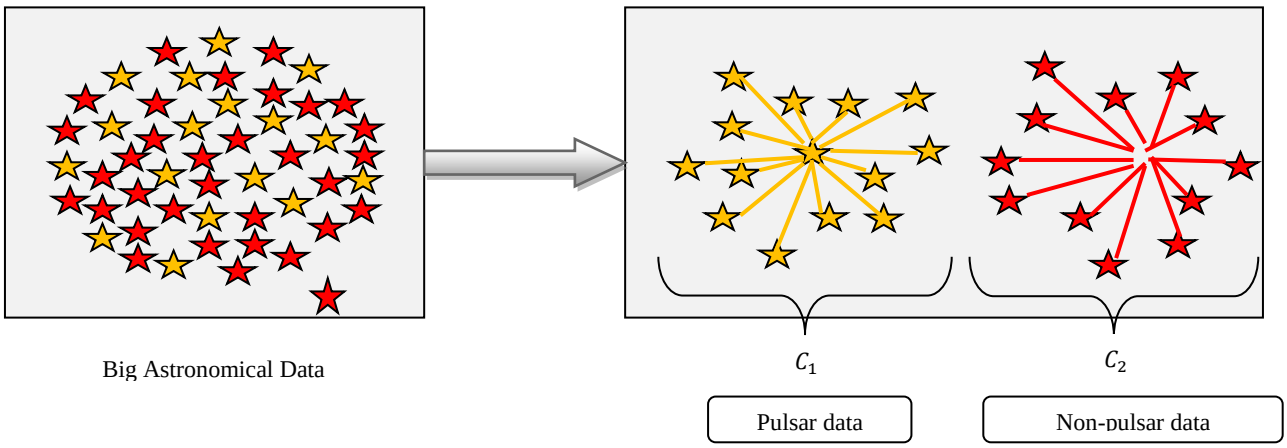

Figure 2. Clustering Results of Big Astronomical Data using GJS-MTAPC Technique

Figure 2 demonstrates clustering results of big astronomical data using proposed GJS-MTAPC technique. For information's available in big astronomical data, GJS-MTAPC technique initializes the two similarity threshold value ranges to form clusters as it contains only two types of data (pulsar data and non-pulsar data). In the case of other types of datasets for different applications, GJS-MTAPC technique set many number of threshold values for efficiently performing big data clustering. The algorithmic processes of GJS-MTAPC Technique is shown in below,

//Generalized Jaccard Similarity based Multilevel Threshold Affinity Propagated Clustering Algorithm
Input: Big Dataset (i.e. Astronomical Dataset)
Output: Enhanced Clustering Accuracy and Reduced Time Complexity For big data
Step 1: Begin
Step 2: Splits big dataset taken as input into a number of subsets using (1)
Step 3: **For** each subset 'SS_i'
Step 4: Randomly select exemplar 'x' for each subset using (2)
Step 5: Calculate similarity between each data samples and exemplar '$S(D_i, x)$' using (3)
Step 6: **End for**
Step 7: **While** (maximum iteration is attained) **do**
Step 8: Update responsibility messages '$r(D_i, x)$' using (4)
Step 9: Update availability messages '$a(D_i, x)$' using (5)
Step 10: Find best exemplars 'X' for all subsets 'SS_i' using (6)
Step 11: **End While**
Step 12: Determine best exemplars and their corresponding similarity value for each subsets
Step 13: Define multiple threshold values

Step 14: Form clusters by grouping every data sample according to similarity values using (7), (8) (9)
Step 15: Return clustering result ' $DS = C_1 \cup C_2 \cup C_3 \ldots \cup C_n$'
Step 16: For

Algorithm 1 Generalized Jaccard Similarity based Multilevel Threshold Affinity Propagated Clustering for Big Data

Algorithm 1 demonstrates the step by step processes of Generalized Jaccard Similarity based Multilevel Threshold Affinity Propagated Clustering (GJS-MTAPC) to group the huge volume of data in dataset. As depicted in algorithm 1, GJS-MTAPC divide the big dataset considered as input into a number of subsets and subsequently choose exemplar for each subset randomly. Then GJS-MTAPC algorithm estimate similarity between each data samples and exemplar. Afterward, the GJS-MTAPC algorithm updates availability and responsibility messages in order to identify the best exemplar in each subset. The similarity value of identified best exemplar is set as threshold values to enhance the clustering performances of GJS-MTAPC algorithm for big data clustering. Finally, GJS-MTAPC algorithm formulates clusters through grouping each data according to similarity value. The algorithmic processes of GJS-MTAPC technique is continued until all the data in big dataset are clustered.

With helps of similarity value between data samples, GJS-MTAPC technique accurately groups the different types of data in a big dataset into number of clusters with minimal false positive rate and minimal time utilization for efficient big data analytics. Therefore, GJS-MTAPC technique attains higher clustering accuracy and lower computational cost to perform big data analytics. Furthermore, GJS-MTAPC technique form clusters by grouping the relevant data only. The irrelevant data are not employed for cluster formation. As a result, GJS-MTAPC technique also minimizes the space complexity of big data analytics.

EXPERIMENTAL SETTINGS

In order to measure the performance, GJS-MTAPC Technique is implemented in Java Language with aid of big astronomical data (i.e. HTRU2 dataset from UCI machine learning repository). The HTRU2 dataset [21] includes of collections of pulsar informations gathered in High Time Resolution Universe (HTRU) survey. The pulsars are a one type of star and also a significant scientific interest. This HTRU2 dataset comprises of 17898 instances. Each candidate is represented by eight continuous variables, and class variable namely mean of the integrated profile, standard deviation of the integrated profile, excess kurtosis of the integrated profile, skewness of the integrated profile, mean of the DM-SNR curve, standard deviation of the DM-SNR curve, excess kurtosis of the DM-SNR curve, skewness of the DM-SNR curve, and class. The first four are statistics acquired from integrated pulse profile. The left over four variables are similarly taken from the DM-SNR curve.

The GJS-MTAPC Technique grouped the numerous data in given big astronomical data into a pulsar and non-pulsar with higher clustering accuracy and minimal computational cost to efficient big astronomical data analytics. The performance of GJS-MTAPC Technique is evaluated in terms of clustering accuracy, computational cost, space complexity and false positive rate. The experimental evaluation of GJS-MTAPC Technique is performed on numerous instances with respect to different number of big data in order to analyze the proposed performances. The efficacy of GJS-MTAPC Technique is compared with existing Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSIO-FCM) [1] and Weighted Consensus Fuzzy Clustering (WCFC) [2] respectively.

RESULTS AND DISCUSSIONS

In this section, performance of proposed GJS-MTAPC Technique is estimated. The result of GJS-MTAPC Technique is compared with existing methods namely, SRSIO-FCM [1] and WCFC [2] with help of metrics such as clustering accuracy, computational cost, space complexity and false positive rate.

Impact of Clustering Accuracy

In GJS-MTAPC Technique, Clustering accuracy computed as ratio of number of data correctly clustered to total number of data samples. The clustering accuracy is estimated in terms of percentage (%) and expressed as,

$$CA = \frac{n_C}{N} * 100 \quad (11)$$

From equation (11), 'N' represents total number of big data taken as input whereas 'n_C' refers number of data correctly clustered. By using equation (11), clustering accuracy 'CA' of GJS-MTAPC Technique is determined with respect to different number of big data. While clustering accuracy of big data analytics is higher, the GJS-MTAPC Technique is said to be more effective.

In order to estimate the clustering performances of big data, GJS-MTAPC Technique is implemented in Java Language by considering different number of data samples in the ranges of 1000-10000 from big astronomical dataset. The performance results of clustering accuracy is compared with existing SRSIO-FCM [1] and WCFC [2] respectively to evaluate efficacy of proposed technique. The GJS-MTAPC Technique attains 89% clustering accuracy whenever taking 5000 data samples from big astronomical dataset as input whereas existing SRSIO-FCM [1] and WCFC [2] obtains 72% and 77 % respectively. Thus, clustering accuracy using GJS-MTAPC Technique is higher when compared to existing SRSIO-FCM [1] and WCFC [2].

The clustering accuracy results obtained during experimental evaluation and clustering performances of GJS-MTAPC Technique is compared with existing two state-of-the-art methods is illustrated in above Table 1. Based on table value, the graph is drawn for analyzing proposed performance which is shown in below Figure 3.

Table 1. Performance Result of Clustering Accuracy for Big Data Analytics

Number of Data	Clustering Accuracy (%)		
	SRSIO-FCM	WCFC	GJS-MTAPC
1000	64	70	84
2000	67	73	85
3000	69	74	87
4000	70	76	88
5000	72	77	89
6000	73	79	90
7000	74	80	92
8000	76	81	93
9000	79	82	95
10000	80	84	96

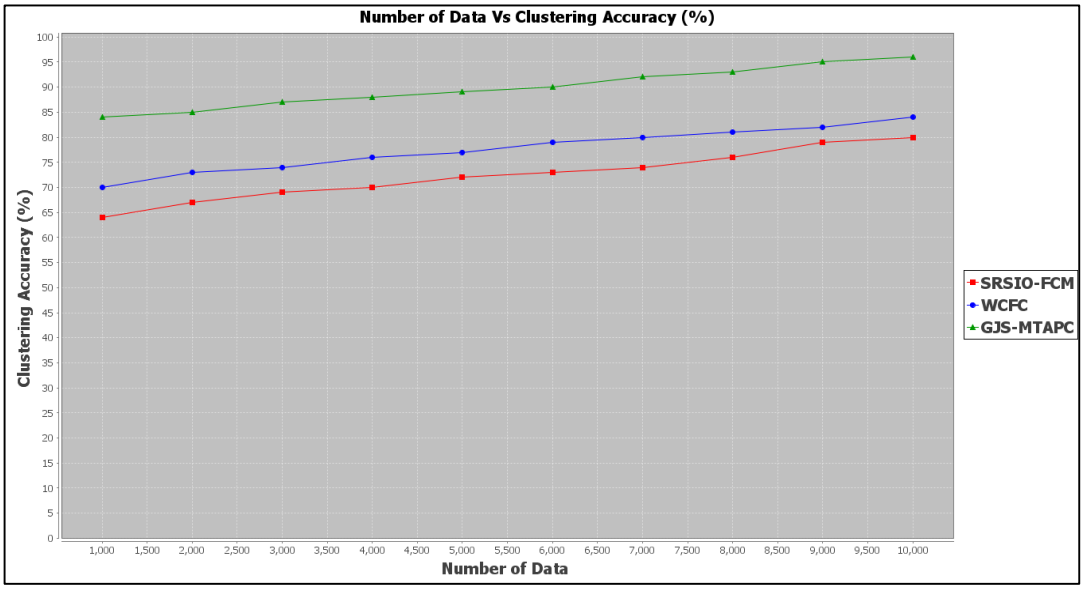

Figure 3. Impact of Clustering Accuracy versus Number of Big Data

Figure 3 depicts the comparative result analysis of clustering accuracy based on diverse numbers of big data in the range of 1000-10000 using three methods namely SRSIO-FCM [1] and WCFC [2] and GJS-MTAPC Technique. As presented in figure, proposed GJS-MTAPC gives higher clustering accuracy for grouping vast amount of data when compared to existing works namely SRSIO-FCM [1] and WCFC [2]. As well while increasing the number of big astronomical data for experimental evaluation, the clustering accuracy is also improved using all the three techniques. But comparatively, clustering accuracy using proposed GJS-MTAPC technique is higher than other existing works. This is owing to application of generalized Jaccard similarity coefficient measurement and multiple threshold values considered in GJS-MTAPC technique.

With support of generalized Jaccard similarity coefficient, GJS-MTAPC technique determines similarity between data samples and exemplars. The higher similarity value indicate the data samples are related whereas low values refer lower similarity value refers data samples are unrelated. With aid of measured similarity values between data samples and exemplars and defined multiple threshold values, GJS-MTAPC technique accurately clusters similar types of data in big dataset. This helps for GJS-MTAPC technique to attain higher clustering accuracy. Hence, proposed GJS-MTAPC technique enhances the clustering accuracy of big data analytics by 24 % and 16 % as compared to existing works namely SRSIO-FCM [1] and WCFC [2] respectively.

Impact of Computational Cost

In GJS-MTAPC Technique, Computational Cost CC estimates the length of time required for big data clustering process. The computational cost is evaluated in terms of milliseconds (ms) and obtained as follows,

$$CC = N * T\ (clustering\ data) \quad (12)$$

From equation (12), the computational cost of big data

analytics is measured. Here, 'N' point outs the number of big data taken as input and 'T' refers amount of time utilized for clustering one data. By using equation (12), computational cost 'CC' of GJS-MTAPC Technique is evaluated with respect to various number of big data. While computational cost of big data analytics is lower, the GJS-MTAPC Technique is said to be more effectual.

The GJS-MTAPC Technique is implemented in Java Languages with aid of various numbers of data samples considered from big astronomical dataset in order to measures the computational cost of big data clustering. The experimental result of computational cost obtained is compared with existing SRSIO-FCM [1] and WCFC [2] respectively to estimate effectiveness of proposed technique. The GJS-MTAPC Technique gets 52 ms computational cost while using 6000 data samples from big astronomical dataset as input for performing experimental processes whereas state-of-the-art works SRSIO-FCM [1] and WCFC [2] acquires 66 ms and 59 ms respectively. From these results, computational cost using GJS-MTAPC Technique is lower when compared to existing SRSIO-FCM [1] and WCFC [2].

The computational cost involved during big data clustering and it experimental results compared with existing SRSIO-FCM [1] and WCFC [2] is portrayed in below Table 2. Depends on table value, the graphical representation is plotted in below Figure 4 for analyzing proposed performance of computational cost for big data analytics

.

Table 2. Performance Result of Computational Cost for Big Data Analytics

Number of Data	Computational Cost (ms)		
	SRSIO-FCM	WCFC	GJS-MTAPC
1000	29	25	18
2000	40	36	27
3000	51	45	39
4000	59	53	46
5000	63	56	50
6000	66	59	52
7000	70	63	56
8000	85	75	68
9000	87	78	71
10000	88	82	73

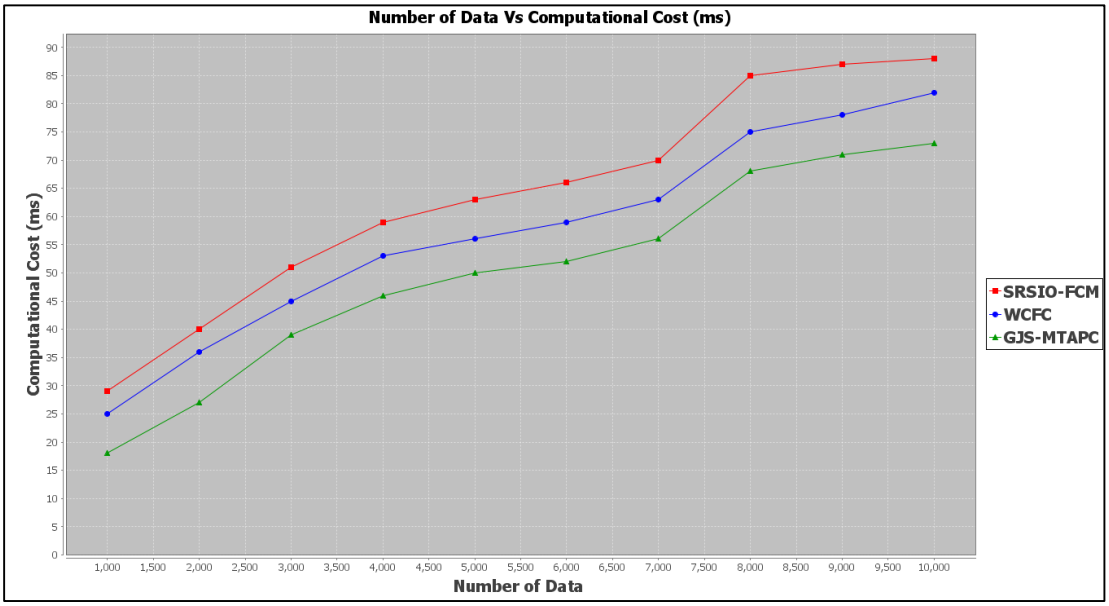

Figure 4. Impact of Computational Cost versus Number of Big Data

Figure 4 shows the experimental result analysis of computational cost with respect to varied numbers of big data in the range of 1000-10000 using three methods namely SRSIO-FCM [1] and WCFC [2] and GJS-MTAPC Technique. As demonstrated in figure, proposed GJS-MTAPC gives lower computational cost results for effective big astronomical

data analytics when compared to existing works namely SRSIO-FCM [1] and WCFC [2]. In addition while increasing the number of big astronomical data for accomplishing experimental processes, the computational cost is also enhanced using all the three techniques. But comparatively, computational cost using proposed GJS-MTAPC technique is lower than other existing works. This is due to application of generalized Jaccard similarity coefficient measurement and multiple threshold values initialized in GJS-MTAPC technique.

By using generalized Jaccard similarity, GJS-MTAPC technique estimates similarity among data samples and exemplars. This assists for GJS-MTAPC technique to identify best exemplars of whole big dataset. Then, GJS-MTAPC technique initializes different threshold values to efficient creation of number of clustering with minimal amount of time utilization. This supports for GJS-MTAPC technique to get lower computational cost. Therefore, proposed GJS-MTAPC technique lessens the computational cost of big data analytics by 23 % and 14 % as compared to existing works namely SRSIO-FCM [1] and WCFC [2] respectively.

Impact of Space Complexity

In GJS-MTAPC Technique, Space complexity SC measures amount of memory space utilized to store the clustered big data. The space complexity is estimated in terms of Mega bytes (MB) and represented as follows,

$$SC = N * memory\ (storing\ data) \quad (13)$$

From equation (13), space complexity needed for big data analytics is determined whereas N indicates number of big data taken as input for experimental evaluation. By using equation (13), space complexity 'SC of GJS-MTAPC Technique is computed with respect to diverse number of big data. While space complexity of big data analytics is lower, the GJS-MTAPC Technique is said to be more efficient.

The experimental evaluation of GJS-MTAPC technique is implemented in java language with aid of dissimilar number of data samples considered from big astronomical dataset to measure the space complexity results of big data analytics. The result of space complexity is compared with existing SRSIO-FCM [1] and WCFC [2] respectively to measure performance of proposed technique. The GJS-MTAPC Technique obtains 53 MB space complexity when utilizing 8000 data samples from big astronomical dataset as input for experimental work whereas existing SRSIO-FCM [1] and WCFC [2] acquires 68 MB and 62 MB respectively. Accordingly, space complexity using GJS-MTAPC Technique is lower when compared to existing SRSIO-FCM [1] and WCFC [2].

Table 3. Performance Result of Space Complexity for Big Data Analytics

Number of Data	Space Complexity (MB)		
	SRSIO-FCM	WCFC	GJS-MTAPC
1000	40	36	28
2000	52	47	39
3000	59	53	45
4000	62	55	47
5000	64	56	48
6000	65	58	50
7000	67	59	51
8000	68	62	53
9000	71	63	55
10000	74	65	59

The space complexity result is determined during big data clustering and it experimental results compared with existing SRSIO-FCM [1] and WCFC [2] is tabulated in below Table 3. With aid of above table values, the graph is formulated in below Figure 6 to find out space complexity result for efficient big data analytics.

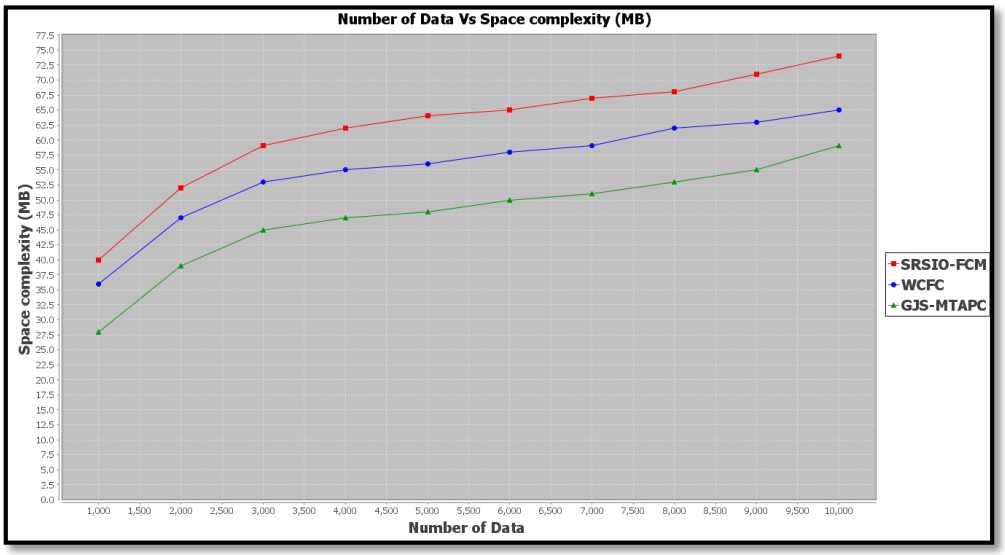

Figure 5. Impact of Space Complexity versus Number of Big Data

Figure 5 illustrates space complexity results is attained during implementation processes based on different numbers of big data in the range of 1000-10000 using three methods namely SRSIO-FCM [1] , WCFC [2] and GJS-MTAPC Technique. As exposed in figure, proposed GJS-MTAPC gives lower space complexity for analytics of big astronomical data when compared to existing works namely SRSIO-FCM [1] and WCFC [2]. Moreover, while increasing the number of big astronomical data during experimental evaluation, the space complexity is also improved using all the three techniques. But comparatively, space complexity using proposed GJS-MTAPC technique is lower than other existing works. This is because the application of generalized Jaccard similarity coefficient and multiple threshold values set in GJS-MTAPC technique.

With computed similarity values between data samples and initialized numerous similarity threshold values, GJS-MTAPC technique correctly clusters the diverse types of data in a big dataset into their corresponding clusters with higher accuracy. Hence, clusters constructed in GJS-MTAPC technique contain only a more related data to that. The unrelated data are eliminated during formation of clusters. Therefore, GJS-MTAPC technique takes minimum amount of memory space to store the clustered big data. This assists for GJS-MTAPC technique to achieve minimum space complexity for big data analytics. Thus, proposed GJS-MTAPC technique minimizes the space complexity of big data analytics by 24 % and 15% as compared to existing works namely SRSIO-FCM [1] and WCFC [2] respectively.

Impact of False Positive Rate

In GJS-MTAPC Technique, False Positive Rate FPR evaluated as ratio of number of data incorrectly clustered to total number of data. The false positive rate of big data clustering is determined in terms of percentage (%) and formulated as,

$$FPR = \frac{n_{IC}}{N} * 100 \qquad (14)$$

From equation (14), 'N' denotes total number of big data considered as input in which 'n_{IC}' indicates number of incorrectly clustered data. By using equation (14), false positive rate 'FPR' of GJS-MTAPC Technique is estimated with respect to dissimilar number of big data. While false positive rate of big data analytics is lower, the GJS-MTAPC Technique is said to be more effectual.

To determine false positive rate of large data clustering, GJS-MTAPC technique considers different number of astronomical data samples in the range of 1000 to 10000 to accomplish experimental work. The result of false positive rate is compared with existing SRSIO-FCM [1] and WCFC [2] respectively to compute performance of proposed technique. The GJS-MTAPC Technique achieves 35 % of false positive rate when taking 4000 data samples from big astronomical dataset as input for experimental processes whereas existing SRSIO-FCM [1] and WCFC [2] acquires 55 % and 46 % respectively. As a result, false positive rate using GJS-MTAPC Technique is lower as compared to existing SRSIO-FCM [1] and WCFC [2].

The false positive rate is measured during processes of big data clustering and it performances compared with existing SRSIO-FCM [1] and WCFC [2] is presented in Table 4. With assist of above shown table values, the graph is plotted in Figure 6 to evaluate false positive rate result for effectual big data analytics.

Table 4. Performance Result of False Positive Rate for Big Data Analytics

Number of Data	False Positive Rate (%)		
	SRSIO-FCM	WCFC	GJS-MTAPC
1000	36	30	16
2000	41	34	21
3000	49	39	28
4000	55	46	35
5000	62	50	41
6000	64	53	45
7000	67	57	47
8000	70	60	50
9000	73	62	52
10000	85	71	61

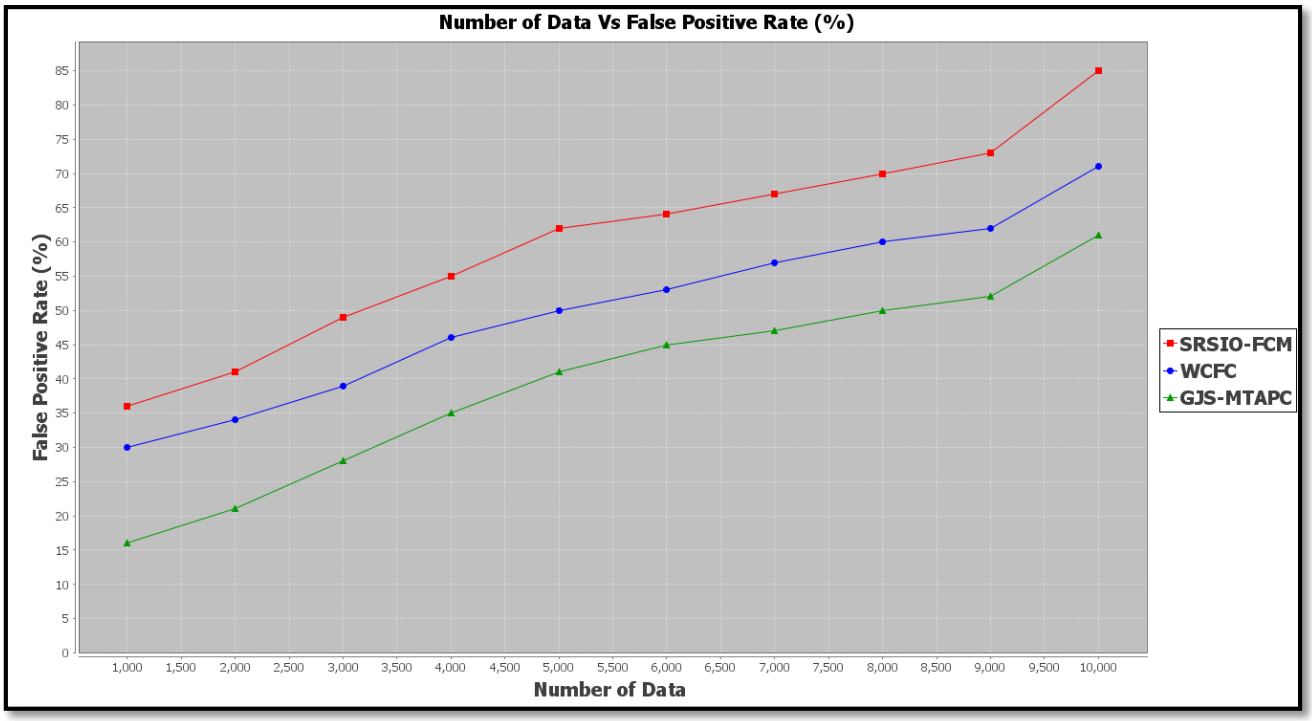

Figure 6. Impact of False Positive Rate versus Number of Big Data

Figure 6 presents experimental result analysis of false positive rate for big data clustering with respect to various numbers of big data in the range of 1000-10000 using three methods namely SRSIO-FCM [1] and WCFC [2] and GJS-MTAPC Technique. As revealed in figure, proposed GJS-MTAPC gives lower false positive rate for clustering massive amount of astronomical data when compared to existing works namely SRSIO-FCM [1] and WCFC [2]. Also, while increasing the number of big astronomical data in order to carry out the experimental process, the false positive rate is also improved using all the three techniques. But comparatively, false positive rate using proposed GJS-MTAPC technique is lower than other existing works. This is due to generalized Jaccard similarity coefficient and diverse ranges of threshold values employed in GJS-MTAPC technique.

By using similarity values among data samples and initialized different similarity threshold values, GJS-MTAPC technique accurately clusters the similar types of data in a big dataset into their corresponding clusters. Hence, clusters formulated in GJS-MTAPC technique include only a more similar data to that. The unwanted or unrelated data are not considered during formation of clusters. Thus, GJS-MTAPC technique lessens number of data that incorrectly clustered for efficient big data analytics. This supports for GJS-MTAPC technique to acquire minimum false positive rate. As a result, proposed GJS-MTAPC technique decreases the false positive rate of big data clustering by 36 % and 23% as compared to existing works namely SRSIO-FCM [1] and WCFC [2] respectively.

CONCLUSION

An effective GJS-MTAPC technique is developed with goal of enhancing AP performances for big data clustering with lesser computational cost. The goal of GJS-MTAPC technique is achieved with applications of Generalized Jaccard Similarity Coefficient measurement and Multilevel Threshold values. The GJS-MTAPC technique significantly groups the various types of data in a very large dataset into a number of clusters with higher precision and lower time. Hence, GJS-MTAPC technique gets enhanced clustering accuracy and decreased false positive rate for efficient big data analytics when compared to state-of-art-works. In addition, GJS-MTAPC technique formulates clusters through grouping only a data samples which is relevant that clusters. Thus, GJS-MTAPC technique lessens the space complexity involved during big data analytics as compared to state-of-art-works. The effectiveness of GJS-MTAPC technique is tested with the metrics such as clustering accuracy, space complexity, computational cost and false positive rate. With the experimental evaluation is performed for GJS-MTAPC technique, it is clear that the clustering accuracy gives more precise results for big data analytics as compared to state-of-the-art works. The experimental results depicts that GJS-MTAPC technique is provides better performance with a improvement of cluster accuracy and the reduction of computational cost of big data clustering when compared to state-of-the-art works.

REFERENCES

[1] 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, Pages 339 – 352, 2016

[2] MinyarSassi Hidri, Mohamed AliZoghlami, RahmaBen Ayed, "Speeding up the large-scale consensus fuzzy clustering for handling Big Data", Fuzzy Sets and Systems, Elsevier, Pages 1-25, 2017

[3] Dheeraj Kumar, James C. Bezdek, Marimuthu Palaniswami, Sutharshan Rajasegarar, Christopher Leckie, Timothy Craig Havens, "A Hybrid Approach to Clustering in Big Data", IEEE Transactions On Cybernetics, Volume: 46, Issue 10 Pages: 2372 – 2385, 2016

[4] Chowdam Sreedhar, Nagulapally Kasiviswanath and 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, Pages 2017

[5] Anan Banharnsakun, "A MapReduce-Based Artificial Bee Colony for Large-Scale Data Clustering", Pattern Recognition Letters, Elsevier, Volume 93, Pages 78-84, July 2017

[6] Mugdha Jain, Chakradhar Verma, "Adapting k-means for Clustering in Big Data", International Journal of Computer Applications, Volume 101, Issue 1, Pages 19-24, September 2014

[7] 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, Pages 1430 – 1445, December 2017

[8] 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, Pages 1–14, 2017

[9] Min Chen, "Soft Clustering for Very Large Data Sets", IJCSNS International Journal of Computer Science and Network Security, Volume 17, Issue 1, Pages 102-108, January 2017

[10] 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, Pages 2445–2452, 2015

[11] 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, Pages 1-8, 2016

[12] 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, Pages 1130 – 1146, 2012

[13] Preeti Arora, Dr. Deepali, Shipra Varshney, "Analysis of K-Means and K-Medoids Algorithm for Big Data", Procedia Computer Science, Elsevier, Volume 78, Pages 507 – 512, 2016

[14] Wenfen Liu, Mao Ye,Jianghong Wei, and Xuexian Hu, "Fast Constrained Spectral Clustering and Cluster Ensemble with Random Projection", Hindawi Computational Intelligence and Neuroscience Volume 2017, Pages 1-4, September 2017

[15] Zhen Liu , Qiuhua Zheng, Zhongping Ji, Weihua Zhao, "Sparse Self-Represented Network Map: A fast representative-based clustering method for large dataset and data stream", Engineering Applications of Artificial Intelligence, Elsevier, Volume 68, Pages 121–130, 2018

[16] Rong Hu , Wanchun Dou, Jianxun Liu, "ClubCF: A Clustering-Based Collaborative Filtering Approach for Big Data Application", IEEE Transactions on Emerging Topics in Computing, Volume 2, Issue 3, Pages 302 – 313, 2014

[17] Panagiotis A. Traganitis, Konstantinos Slavakis, Georgios B. Giannakis, "Sketch and Validate for Big Data Clustering", IEEE Journal of Selected Topics in Signal Processing, Volume 9, Issue 4, Pages 678 – 690, 2015

[18] Minchao Wang, Wu Zhang, Wang Ding, Dongbo Dai, Huiran Zhang, Hao Xie, Luonan Chen, Yike Guo, Jiang Xie, "Parallel Clustering Algorithm for Large-Scale Biological Data Sets", PLoS ONE, Volume 9, Issue 4, Pages 1-9, 2014

[19] Amin Mohebi, Saeed Aghabozorgi, Teh Ying Wah, Tutut Herawan and Ramin Yahyapour, "Iterative big data clustering algorithms: a review", Wiley Online Laboratory, Volume 46, Issue 1, Pages 107–129, January 2016

[20] Preeti Gupta, Arun Sharma, and Rajni Jindal, "Scalable machine-learning algorithms for big data analytics: a comprehensive review", Wiley Online Laboratory, Volume 6, Issue 6, Pages 194–214, 2016

[21] HTRU2 Data Set: https://archive.ics.uci.edu/ml/datasets/HTRU2