BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 02/08 pp240–247
Volume 2, Number 4, December 2019
DOI: 10.26599/BDMA.2018.9020037

A Novel Clustering Technique for Efficient Clustering of Big Data in Hadoop Ecosystem

Sunil Kumar* and Maninder Singh

Abstract: Big data analytics and data mining are techniques used to analyze data and to extract hidden information. Traditional approaches to analysis and extraction do not work well for big data because this data is complex and of very high volume. A major data mining technique known as data clustering groups the data into clusters and makes it easy to extract information from these clusters. However, existing clustering algorithms, such as k-means and hierarchical, are not efficient as the quality of the clusters they produce is compromised. Therefore, there is a need to design an efficient and highly scalable clustering algorithm. In this paper, we put forward a new clustering algorithm called hybrid clustering in order to overcome the disadvantages of existing clustering algorithms. We compare the new hybrid algorithm with existing algorithms on the bases of precision, recall, F-measure, execution time, and accuracy of results. From the experimental results, it is clear that the proposed hybrid clustering algorithm is more accurate, and has better precision, recall, and F-measure values.

Key words: clustering; Hadoop; big data; k-means; hierarchical

1 Introduction

Big data is currently generating a buzz in the market and data is rapidly growing from being measured in gigabytes to terabytes, petabytes, and zetabytes[1]. Big data has such large data requirements that applications that were previously used to store and process data—Database Management System (DBMS), Relational Database Management System (RDBMS), etc.—are now failing the data demand[2]. Big data includes extremely large datasets, meaning that it is not possible for commonly used software tools to manage and process that data within the required time frame[3]. Therefore, massively parallel software running across many servers is now required to handle this workload[4]. Big data requires techniques to reveal insights from datasets that are diverse, complex, and of a massive scale. Some of the challenges of big data processing include difficulties in data capture, meeting the need for speed, addressing data quality, dealing with outliers, sharing of big data, and big data analysis[5]. A number of techniques have been proposed to data in order to handle big data datasets, e.g., machine learning, association techniques, support vector machines, and clustering[6]. In this paper, we propose a new hybrid clustering technique to handle big data.

The rest of this paper is organized as follows. Existing clustering is explained and their advantages outlined in Section 2. The proposed hybrid clustering technique is explained in Section 3. Section 4 includes the experimental setup and the results of experiments using the proposed algorithm are explained in Section 5. Finally, in Section 6, we summarize our results and present suggestions for future work.

• Sunil Kumar is with the Directorate of Livestock Farms, Guru Angad Dev Veterinary and Animal Sciences University, Ludhiana 141001, India. E-mail: sunilkapoorldh@gmail.com.
• Maninder Singh is with the Department of Computer Science, Punjabi University, Punjab 147002, India. E-mail: singhmaninder25@yahoo.com.
* To whom correspondence should be addressed.
Manuscript received: 2018-11-08; revised: 2019-01-09; accepted: 2019-01-12

2 Clustering Analysis Techniques

Cluster analysis is a data mining task that aims to provide for search, recommendation, and organization of data. In clustering techniques, datasets are grouped into a number of clusters each with different attributes[7,8]. Clustering is in a class of unsupervised learning techniques, unlike classification, in which similar objects of the dataset are grouped into clusters[9], and thus form different clusters such that objects in the same cluster groups are very different from each other and objects in the same group or cluster are very similar to each other[10,11]. The clusters are known only after the complete execution of the clustering algorithm[12]. Two clustering algorithms that are used for managing large datasets are k-means clustering and hierarchical clustering, each of which is summarized below.

2.1 *k*-means clustering

k-means is a partition-based clustering method that is unsupervised, non-deterministic, numerical, and iterative[13]. In this method, n objects are partitioned into k clusters such that there should be low inter-cluster similarity and high intra-cluster similarity[14]. Every cluster has a centroid, or cluster representative, from which the distance of all data points is measured and data points at the minimum distance from the centroid are kept under one cluster[15,16]. This algorithm explores the structure of a dataset. To apply the k-means algorithm, the number of clusters needs to be predefined and randomly chosen k points may serve as initial centroids[17,18]. The red dots in Fig. 1 represent one cluster with a centroid at (0, 0) whereas the black dots represent another cluster centered at (1,1).

k-means clustering has the following limitations:

- The k-means algorithm is a static algorithm;
- It is hard to predict the k value;
- k-means clustering requires a predefined number of clusters;
- The initial centroid is chosen randomly;
- It is sensitive to outliers and these are difficult to detect;
- k-means clustering produces single partitioning;
- The clusters formed are not of good quality;
- k-means clustering forms clusters of a fixed shape, i.e., convex;
- k-means clustering is sensitive to noisy data;
- Many data points do not fit into any of the clusters; and
- Compared to other algorithms, accuracy and F-measure are lower.

2.2 Hierarchical clustering

Hierarchical clustering is a nested approach in which one cluster is nested into another, thus forming a sequence or proper hierarchy[19]. Hierarchical clustering combines small clusters into large clusters, and can also divide or split a larger cluster into smaller ones[20]. Figure 2 shows the example of hierarchical clustering, i.e., one approach is further divided in the tasks. The result of the algorithm is a tree of clusters showing how the clusters are related. It is used in data mining tasks to analyze big data[21]. The two main types of hierarchical clustering are agglomerative and divisive.

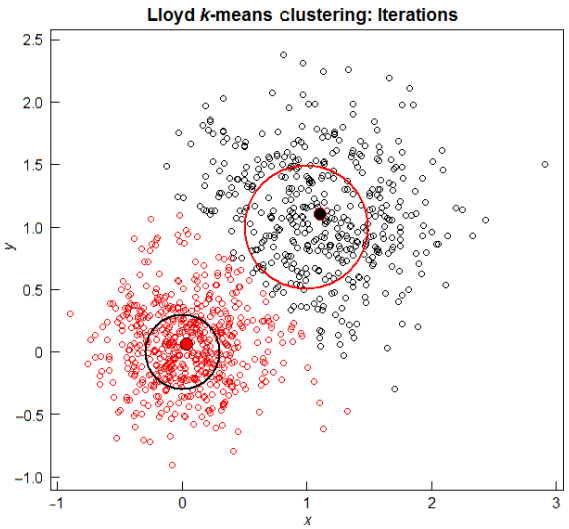

Fig. 1 *k*-means partition clustering.

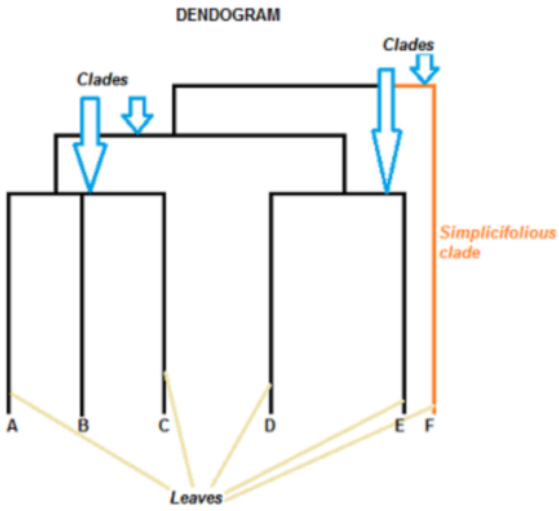

Fig. 2 Hierarchical clustering.

Agglomerative clustering executes from bottom to top. Each data point is treated as a single object and then combines with other objects to form a cluster[22]. These clusters are then combined successively until a single big cluster is obtained. Agglomerative algorithm has a time complexity of $O(n^2)$[21]. Where clusters are of equal volume, the complete-link method is the best option.

Divisive clustering places all the objects in one cluster initially and then successive splits are done to separate clusters[23]. These iterations are performed until the desired number of clusters is obtained.

Hierarchical clustering has the following limitations:

- It is not possible to represent distinct clusters with the same kind of expression patterns;
- As clusters grow in size, the actual expression patterns become less relevant;
- The complexity is quadratic;
- It is less efficient than k-means clustering in run time;
- It has to make several merge split decisions;
- It takes more time to execute than the k-means algorithm; and
- Many iterations are involved in hierarchical clustering.

3 Proposed Hybrid Clustering Technique

In this paper, we propose a new hybrid clustering technique that combines the workings of earlier clustering algorithms. This new approach combines the functionality of the two above mentioned techniques while eliminating their disadvantages, leading to better results in data output.

The algorithm steps of the hybrid approach is shown as follows:

(1) Read the data set file using the Java Buffer class.

(2) Apply k-means clustering. Find clusters based on the k-means algorithm.

(3) On receiving the rules in the Mapper class, Mapper filters the data.

(4) The clusters formed by k-means clustering are not adequate and many data points still do not lie in any of the clusters, and the quality of clusters is also poor. So check if the required number of clusters are formed, and check if all data points have been covered.

(5) If not, then apply the hierarchical algorithm. Skip the pass in which the clusters have already been formed by k-means algorithm, thus saving iterations of the hierarchical algorithm. Hierarchical algorithm makes new clusters by its own algorithm.

(6) Store all points to hash map now to remove redundancy in clusters formed. Mapper fetches the data points of both the algorithms and passes to the Reducer class. Combining the clusters of both the algorithms enhances the performance of the hybrid algorithm.

4 Experimental Setup

To implement the proposed hybrid clustering technique in Hadoop[24], we chose a dataset of the National Climatic Data Center (NCDC), containing the world's largest active archive of weather data[25]. It contains weather files that are constructed in standard ASCII format and is globally available to everyone. This global database integrates the surface hourly data from 20 000 stations all over the world.

The NCDC dataset has weather files for different years starting from 1901. Weather is recorded on every day of a year. The input dataset of a weather file selected for a particular year looks as shown in the snapshot, which shows the weather file for year 1907. A brief description of each of the 32 attributes is given in Fig. 3.

The dataset consists of 3 sections: a control section, a mandatory data section, and an additional data section; these are described below.

Every record starts with a fixed length **control section** of 60 characters. The control section contains information about the report data such as the observation date, time, station location information, etc. A brief introduction to every attribute in the control section is provided in Table 1.

The control section is followed by a **mandatory section,** also of a fixed length, of 45 characters long. This section contains meteorological information about the temperature, pressure, winds, etc. A brief introduction to every attribute in the mandatory section is also provided in Table 1.

After the mandatory section comes an additional data section, which is not of fixed length and contains a variable number of characters. It is not mandatory so there need only be two sections (control and mandatory) of a given record. Sometimes a remark or element quality section can be included after the additional data section.

The proposed algorithm follows two major stages in MapReduce parallel processing. The output of the map phase is fed as input to reduce.

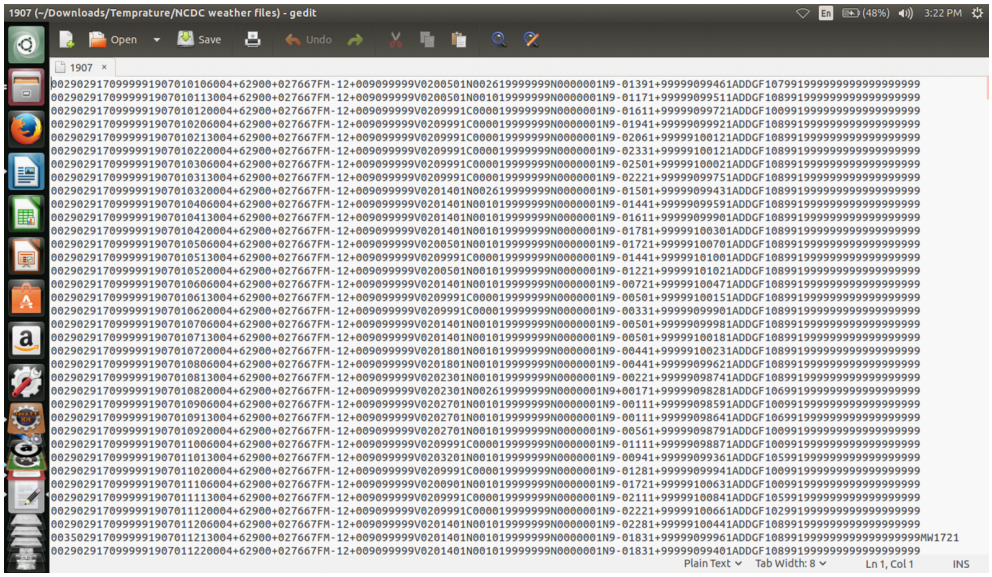

Fig. 3 Weather file of year 1907.

Table 1 Attributes in dataset file.

Position	Attribute No.	Attribute name	Length of attribute (Character)
1–4	1	Total-variable characters	4
5–10	2	Fixed-weather station USAF master station catalog identifier	5
11–15	3	Fixed-weather station National Centers for Environmental Information (NCEI) Weather Bureau Army Navy (WBAN) identifier	4
16–23	4	Geophysical point observation date	7
24–27	5	Geophysical point observation time	4
28	6	Geophysical point observation data source flag	1
29–34	7	Geophysical point observation latitude coordinate	5
35–41	8	Geophysical point observation longitude coordinate	6
42–46	9	Geophysical report type code	6
47–51	10	Geophysical point observation elevation dimension	5
52–56	11	Fixed weather station call letter identifier	5
57–60	12	Meteorological point observation quality control process name	4
61–63	13	Wind observation direction angle	3
64	14	Wind observation direction quality code	1
65	15	Wind observation type code	1
66–69	16	Wind observation speed rate	4
70	17	Wind observation speed quality code	1
76	18	SKY condition observation ceiling quality code	1
77	19	SKY condition observation ceiling determination code	1
78	20	SKY condition observation CAVOK code	1
79–84	21	Visibility observation distance dimension	6
85	22	Visibility observation distance quality code	1
86	23	Visibility observation variability code	1
87	24	Visibility observation distance quality variability code	1
88–92	25	Air temperature observation air temperature	5
93	26	Air temperature observation air temperature quality code	1
94–98	27	Air temperature observation dew point temperature	5
99	28	Air temperature observation dew point quality code	1
100–104	29	Atmospheric pressure observation sea level pressure	5
105	30	Atmospheric pressure observation sea level pressure quality code	1

Mapper stage. The input to the Mapper phase is the weather file for a selected year, as per the example of 1907 given in Fig. 3. Here we extract two fields from every record: the observation date and the air temperature. Also the quality code is tested to check that its value is not missing. As the MapReduce paradigm is based on a key-value scenario, so here also the Mapper divides the extracted data into key-value pairs. We pass the date (as key) and temperature (as value) to the reducer phase, with the temperature divided by 10. Here the key is in Text form and the value is in IntWritable form.

The key-value pairs for this dataset are in the following form: (Observation date, Air temperature (Celsius)); some examples are (19070101, 13.9), (19070102, 11.7), and (19070103, 12.3). Every key-value pair is unique. The two fields extracted in the Mapper stage—observation date and observed air temperature on that date—are shown as an output file in Fig. 4.

Reduce stage. The reduce phase accepts the output of the Mapper phase as its input. It receives the key-value pairs in Text and IntWritable form, respectively. That is, all the temperature values belong to a particular observation date at a specified time and in specific conditions. The temperature may be measured more than once but up to a maximum of three times on a particular day, so every key value is specific. The Reducer's performance varies for different algorithms for making clusters and finding the maximum temperature from the clusters formed.

5 Experimental Results

The proposed hybrid clustering technique was compared with the existing k-means and hierarchical algorithms. These were compared on precision, recall, F-measure, execution time, and number of clusters formed. The results on these comparisons are as follows.

5.1 Precision

Precision is measured as the fraction of pairs of data points correctly placed into the same cluster. It is directly proportional to the quality of clusters formed and accuracy; the lower is the precision, the poorer is the quality of clusters formed; the greater is the precision, the more accurate is the algorithm and the higher is the quality of clusters formed.

Precision=Clusters computed by particular algorithm÷
Actual clusters in the dataset.

It is measured as the deviation from the actual value. For the clustering algorithms, precision is measured as the clusters formed by a particular algorithm divided by the actual clusters that can be formed in the dataset. Alternatively, we could describe it as the percentage of relevant clusters returned by the algorithm. From Fig. 5, it is clear that the hybrid algorithm has the highest precision whereas the k-means algorithm has the lowest.

5.2 Recall

Recall is also measured on a pair of data points; it is the fraction of the actual pairs of data points that were

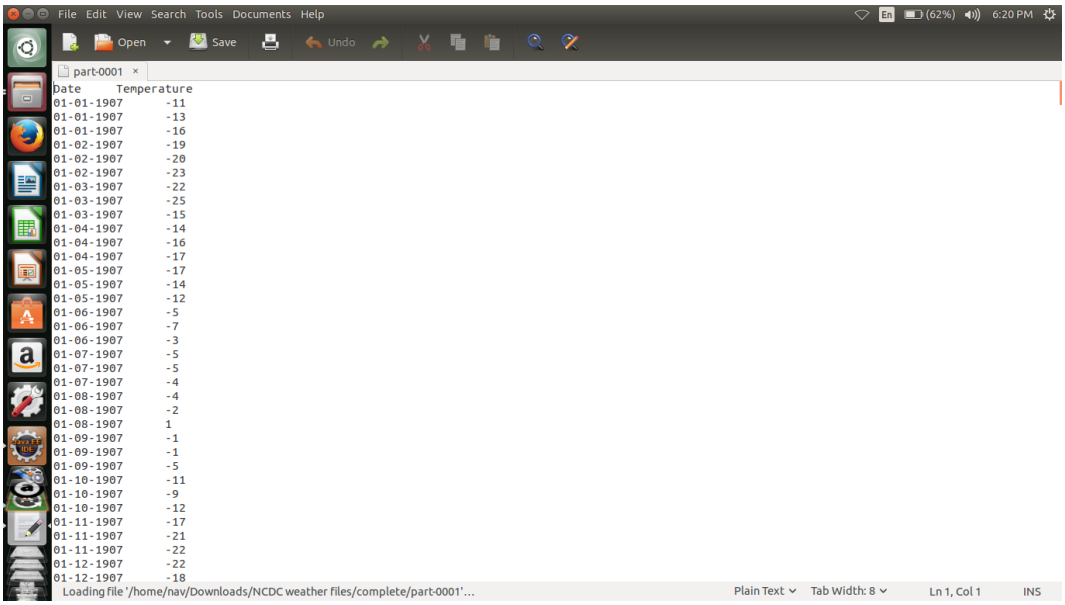

Fig. 4 Output of Mapper stage in a file.

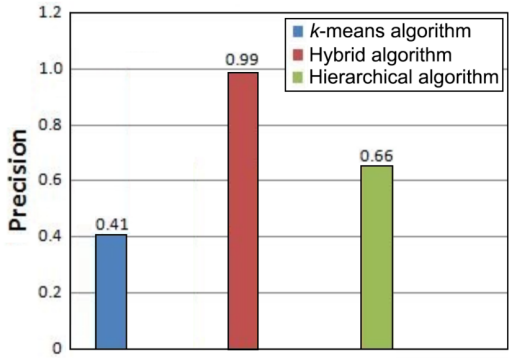

Fig. 5 Comparison of three algorithms based on precision.

identified. Recall is also directly proportional to cluster quality.

From Fig. 6, it is clear that the hybrid algorithm has the highest recall whereas the k-means algorithm has the lowest.

5.3 F-measure

The F-measure calculation is directly proportional to precision and recall; the greater is the precision and recall, the greater is the F-measure. F-measure is calculated as

$$(2\times\text{precision}\times\text{recall})/(\text{precision} + \text{recall}).$$

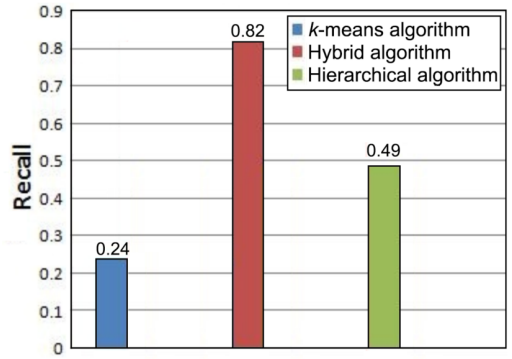

Fig. 6 Comparison of three algorithms based on recall.

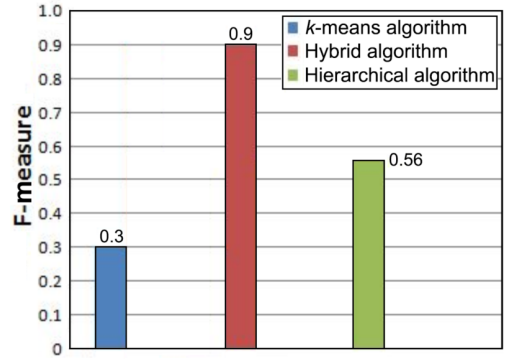

Fig. 7 Comparison of three algorithms based on F-measure.

From Fig. 7, it is clear that the hybrid algorithm has the highest F-measure, thus a greater accuracy of clusters formed, whereas the k-means algorithm has the lowest.

5.4 Execution time

All the three clustering algorithms are compared on the time factor. Figure 8 compares the execution time of all three algorithms. Here the k-means algorithm takes the least time to make clusters, whereas the hybrid algorithm requires the most.

5.5 Number of clusters formed

The k-means algorithm covers less data points and many outliers are detected as shown in Fig. 9. Many data points are left outside of any cluster.

The hierarchical algorithm is dynamic in nature and makes clusters at run time. The number of clusters need not be predefined. It includes a greater number of data points under clusters, and generates a greater number of clusters as shown in Fig. 10.

The hybrid algorithm covers almost every data point and places these data points with other that are similar

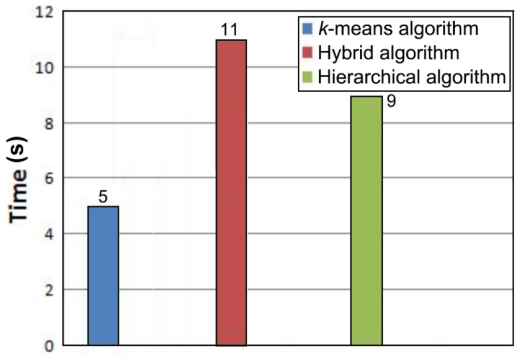

Fig. 8 Comparison of three algorithms based on execution time.

Month	Count
02-1907	84
04-1907	90
06-1907	90
09-1907	90
11-1907	90

Fig. 9 Data points count of k-means approach.

Month	Count
01-1907	93
02-1907	84
03-1907	93
05-1907	93
07-1907	93
08-1907	93
10-1907	93
12-1907	93

Fig. 10 Data points count of hierarchical approach.

in nature. It generates more clusters and the clusters formed are of very good quality as shown in Fig. 11.

All the final results are mentioned in Table 2 and hybrid approach has the best results as compared to existing algorithms.

6 Conclusion

In this paper, we have implemented the proposed hybrid clustering approach with the MapReduce under Hadoop. Dealing with big data, we compare clustering algorithms using NCDC weather data files. We set out to find the day of a selected year with the maximum temperature, by making clusters with various clustering algorithms. Each algorithm has some drawbacks; k-means generates only a few clusters and also requires predefining of the number of clusters to be formed as it is static in nature, whereas hierarchical clustering is dynamic in nature and generates more clusters than k-means but runs many iterations due to the need for many merge and split decisions. Due to these problems, we combined both algorithms to realize the merits of both while discarding their demerits. With the resulting hybrid approach, we find the maximum number of clusters from a file, and the clusters formed are of very good quality, thus producing most accurate results. The proposed hybrid approach produces an efficient way of clustering showing higher precision, recall, and F-measure. The result produced by the efficient hybrid clustering algorithm is most accurate, as the calculated maximum temperature value is the actual maximum temperature value. The hybrid algorithm produces the highest number of clusters and includes every data point in any one of these clusters. Also, the Map output records are maximized for this algorithm in MapReduce framework.

Month	Count
01-1907	93
02-1907	84
03-1907	93
04-1907	90
05-1907	93
06-1907	90
07-1907	93
08-1907	93
09-1907	90
10-1907	93
11-1907	90
12-1907	93

Fig. 11 Data points count of hybrid approach.

The proposed technique for the efficient clustering of big data improves on other clustering approaches in many respects, but the hybrid clustering approach has the limitation of taking more time to execute than either of the k-means and hierarchical clustering methods. Future research is needed to reduce its execution time using different parameters.

References

[1] C. L. P. Chen and C. Y. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on big data, *Inf. Sci.*, vol. 275, pp. 314–347, 2014.

[2] A. Gandomi and M. Haider, Beyond the hype: Big data concepts, methods, and analytics, *Int. J. Inf. Manage.*, vol. 35, no. 2, pp. 137–144, 2015.

[3] W. Raghupathi and V. Raghupathi, Big data analytics in healthcare: Promise and potential, *Health Inf. Sci. Syst.,* vol. 2, p. 3, 2014.

[4] B. Saraladevi, N. Pazhaniraja, P. V. Paul, M. S. S. Basha, and P. Dhavachelvan, Big data and Hadoop-A study in security perspective, *Procedia Computer Science*, vol. 50, pp. 596–601, 2015.

[5] A. Katal, M. Wazid, and R. H. Goudar, Big data: Issues, challenges, tools and good practices, in *Proc. 6th Int. Conf. Contemporary Computing*, Noida, India, 2013, pp. 404–409.

[6] M. Herland, T. M. Khoshgoftaar, and R. Wald, A review of data mining using big data in health informatics, *J. Big Data*, vol. 1, p. 2, 2014.

[7] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, A survey of clustering algorithms for Big Data: Taxonomy and empirical analysis, *IEEE Trans. Emerg. Top. Comput.*, vol. 2, no. 3, pp. 267–279, 2014.

[8] X. B. Li and Z. X. Fang, Parallel clustering algorithms, *Parallel Comput.*, vol. 11, no. 3, pp. 275–290, 1989.

[9] J. Dittrich and J. A. Quiane-Ruiz, Efficient big data processing in Hadoop MapReduce, *Proceedings of the VLDB Endowment,* vol. 5, no. 12, pp. 2014–2015, 2011.

[10] C. C. Aggarwal and C. X. Zhai, A survey of text clustering algorithms, in *Mining Text Data*, C. C. Aggarwal and C. X. Zhai, eds. Springer, 2012, pp. 77–128.

[11] A. Hatamlou, In search of optimal centroids on data clustering using a binary search algorithm, *Pattern Recognit. Lett.*, vol. 33, no. 13, pp. 1756–1760, 2012.

[12] D. Pandove and S. Goel, A comprehensive study on clustering approaches for big data mining, in *Proc. 2nd Int. Conf. Electronics and Communication System*, Coimbatore, India, 2015, pp. 1333–1338.

Table 2 Final result analysis.

Algorithm	TP rate	FP rate	Precision	Recall	F-measure	Execution time (s)	Data point
Hybrid clustering approach	1.2	0.027	0.99	0.82	0.90	11	12
k-means clustering	1.00	0.037	0.41	0.24	0.30	5	5
Hierarchical clustering	0.901	0.072	0.66	0.49	0.56	9	8

[13] R. Jensi and G. W. Jiji, Hybrid data clustering approach using k-means and flower pollination algorithm, *Adv. Comput. Intell.: Int. J.*, vol. 2, no. 2, pp. 15–25, 2015.

[14] B. B. Ali and Y. Massmoudi, K-means clustering based on Gower Similarity Coefficient: A comparative study, in *Proc. 5^{th} Int. Conf. Modeling, Simulation and Applied Optimization*, Hammamet, Tunisia, 2013.

[15] A. Hatamlou, S. Abdullah, and H. Nezamabadi-Pour, A combined approach for clustering based on k-means and gravitational search algorithms, *Swarm Evol. Comput.*, vol. 6, pp. 47–52, 2012.

[16] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, 2002.

[17] B. B. Firouzi, M. S. Sadeghi, and T. Niknam, A new hybrid algorithm based on PSO, SA, and K-means for cluster analysis, *Int. J. Innova. Comput., Inf. Control*, vol. 6, no. 7, pp. 3177–3192, 2010.

[18] Y. K. Patil and V. S. Nandedkar, Design and implementation of k-means and hierarchical document clustering on hadoop, *Int. J. Sci. Res.*, vol. 3, no. 10, pp. 1566–1570, 2014.

[19] E. Rashedi and A. Mirzaei, A novel multi-clustering method for hierarchical clusterings based on boosting, in *Proc. 9^{th} Iranian Conf. Electrical Engineering*, 2011, pp. 1–5.

[20] R. T. Ng and J. W. Han, CLARANS: A method for clustering objects for spatial data mining, *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, 2002.

[21] A. Farinelli, M. Bicego, S. Ramchurn, and M. Zucchelli, C-link: A hierarchical clustering approach to large-scale near-optimal coalition formation, in *Proc. 23^{rd} Int. Joint Conf. Artificial Intelligence*, Beijing, China, 2013, pp. 106–112.

[22] A. Mirzaei and M. Rahmati, A novel hierarchical-clustering-combination scheme based on fuzzy-similarity relations, *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 27–39, 2010.

[23] E. M. Rasmussen and P. Willett, Efficiency of hierarchic agglomerative clustering using the ICL distributed array processor, *J. Doc.,* vol. 45, no. 1, pp. 1–24, 1989.

[24] Apache Hadoop, http://hadoop.apache.org/, 2018.

[25] National Climatic Data Centre (NCDC) Data Access, https://www.ncdc.noaa.gov/data-access, 2018.

Sunil Kumar received the MS degree from Punjabi University, India in 2010. He is working in the Directorate of Livestock Farms, Guru Angad Dev Veterinary and Animal Sciences University, India. He is also pursuing the PhD degree at Punjabi University, India. His research interests include wireless networks, ad-hoc networks, and big data analytics.

Maninder Singh is working as an assistant professor in the Department of Computer Science, Punjabi University, India. He received the PhD degree from Punjabi University, India in 2009. He is the member of various professional societies such as IEEE, ACM, CSTA, etc. His research interests include wireless networks, ad-hoc networks, big data analytics, and machine learning.