JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Implementation of Clustering Algorithms for Massive Amount of Date

S Archana[1], **Dr.Neeraj Sharma** [2], **Dr.Pradosh Chandra Patnaik**[3]

[1]*Research Scholar, Dept. of Computer Science and Engineering*
Sri Satya Sai University of Technology and Medical Sciences,
Sehore Bhopal-Indore Road, Madhya Pradesh, India.

[2]*Research Guide, Dept. of Computer Science and Engineering*
Sri Satya Sai University of Technology and Medical Sciences,
Sehore Bhopal-Indore Road, Madhya Pradesh, India.

[3]*Research Co-Guide, Professor & Principal . Dept. of Computer Science and Engineering*
Aurora's PG College (MCA) Hyderabad

Abstract

The exploratory nature of data analysis and data mining makes clustering one of the most usual tasks in many applications like biology, text analysis, signal analysis, etc that involve huge amount of datasets . Traditional Clustering methods like K-means or hierarchical clustering are beginning to reach its maximum capability to cope with this increase of dataset size. The limitation for these algorithms come either from the need of storing all the huge data in memory or because of their computational time complexity. These have been opened an area for research of algorithms that able to reduce this overhead. In one perspective the solutions can be in the stage of data pre-processing by transforming the data to a lower dimensionality manifold that represents the structure of the data or at the last stage of summarizing the dataset by obtaining a smaller subset of examples that represent an equivalent information. A Second perspective is to modify the Traditional clustering algorithms or to derive other ones that are able to cluster larger datasets. This perspective depends on many different approaches. An Approaches such as sampling techniques, on-line processing, summarization, and efficient data structures have being applied to the problem of scaling clustering algorithms. This paper presents a review of different approaches and clustering algorithms that apply these techniques. The aim is to cover various methodologies applied for clustering data and how they can be scaled.

Introduction

Now a days, data is being produced at a very fast rate and resulting into a huge mass of data that is referred to as Big Data. Big data exhibits various features such as huge volume, diverse variety, highly varying, multi-valued, varying swiftness and huge convolution ,which makes it difficult to examine the data and obtain the required information with traditional data mining methods. Extracting meaningful and required information or to find out unrelated relationship between the data is viewed as Data mining. There are different stages that data mining technology has come across, which assistances the business organizations to nurture their business figures or helps them in decision making. Big data have five different stages of data mining. In Stage one we make use of a solitary machine for a solitary algorithm that uses vector data, in second stage we go for numerous algorithms used for database combinations, third stage is used where grid computing is used and the distribution of data is in the fourth stage. Finally ,Parallel data mining algorithms are used in the Last stage. The parallel data mining method can be partitioned into four modules such as classification algorithms to classify the data, association rule mining to generate rules, clustering algorithms to differentiate the data without any training data set given and stream data mining algorithms. Sparse datasets in bigdata is the major challenge for designing and developing efficient techniques for mining partial and inexact data. Because of variety of applications , analysis of clustering data has been an evolving problem in datamining.These algorithms have become popular because of development of various data clustering tools and their comprehensive use in a a broad range of applications like image processing,computational biology, medicine,mobile communication and economics which inturn improves the popularity of these clustering algorithms.Standerization of clustering algorithms is the main issue.The results may be performing best for one type of dataset ,but may perform poorly for other type of datasets .Though efforts were made to standardize the algorithms which performs well in any kind of situations, no significant progress is made.Since,every algorithm has its own pro's and con's a single algorithm cannot be designed which can be used for all sorts of data.

The ability of the algorithm to handle a growing amount of input by adding additional resources to the system is measured in terms of its scalability. Based on the size of the work ,the system can be scaled up or down. In todays world, the scaling the resources has become an essential factor as a result of the cost of adding resources to the system.That's why research is carried out in these areas for developing ways to deal with scalable systems, especially in cases of big data and what means real-time versus cost. Based on Scalabilty , the techniques are broadly divided into two types: traditional and scalable techniques. The traditional techniques consist of clustering algorithms without regard to the system's scalability. In contrast, the scalable techniques consist of the clustering algorithms which take into account the system's scalability. Due to the limitations of the traditional clustering algorithms either in output speed or in processing data, researchers investigated in two directions to face these challenges. The first direction is by trying to improve traditional algorithms to working with large data size and the other orientation by proposed new methodology based

on the benefit of new technology such as parallel computing, cloud computing, and map-reduce.

REVIEW OF THE LITERATURE

Bozdemir et al., (2021) On education datasets, a parallel clustering technique based on message passing interface (MPI) called M-K-Means is applied. M-K-Means is composedof MPI and Sequential K-Means. Additionally, the K-Mean and Message Passing interfaces are employed concurrently in the same experiment, which is likewise based on random centroids selection and divides a dataset into "p" sub-datasets, where "p" is the number of nodes connected to the main computer. The approach is validated on a DNA dataset and the Simple K-Mean algorithm is parallelized with randomly chosen initial centroids. When the data set is tiny, the classic clustering technique performs well and produces satisfactory results. However, it struggles with huge datasets due to a variety of reasons, including data volume, data dimensionality, computational power, and memory. Numerous strategies have been developed to increase the efficiency and accuracy ofclustering results in a large amount of data environment.

Wang et al., (2021) conducted a similar analysis and discovered that a threshold value is used to create this new list. When the new list's items meet the threshold, the new list's values are returned as initial centroids. The Parallel K-Mean clustering algorithm is implemented using the Para Means programme. They parallelize the Simple K-Mean clustering technique for common laboratory application. Para Means is a client-server application that is simple to manage. Cluster analysis has been extensively studied as a subfield of statistics for many years. Numerous statistical analysis software packages or systems now include analysis tools. Multiple dimensions or attributes may exist in a database. Numerous clustering techniques are efficient in terms of processing. Generally, two to three dimensions are involved in low-dimensional data. In general, the quality of clustering can be accurately assessed only in three-dimensional scenarios. Clustering data items in a high-dimensional space is extremely difficult, much more so when the data is severely skewed and sparse. Capacity for noisy data processing: The majority of data in practical applications include outliers, such as missing, unknown, or inaccurate data.Certain clustering methods are sensitive to this type of data, resulting in low-quality clustering results.

Karwan Qader et al. (2017) investigated three distinct data mining techniques, including K-Means, Fuzzy C Means, and EM, as well as a 44-class strategy for network fault classification. The established approach benefited in identifying anomalous behaviors in communication networks and provided a means of real-time fault classification and management. Then, datasets from networks with high and low traffic volumes were collected, and a prototype was created for performing network traffic fault classification under the provided conditions. k-Means and EM algorithms significantly reduced the processing overhead compared to other standard techniques. However, the time complexity remained over the acceptable level. By optimising the findings of network and system measurements, the similarity measure

addressed the shortcomings of existing techniques. Additionally, the similarity measure aided in traffic clustering. However, itwas unable to forecast traffic flows.

Yu Wang et al. (2016) devised a limited clustering approach to improve traffic clustering accuracy. The decisions were constructed using a constrained clustering approach in conjunction with background traffic statistics. Additionally, a set of similar constraints was included in the constrained clustering technique. Then, to maximise the evaluation of algorithm parameters, we employed Gaussian mixture density and an approximate approach called the SBCK algorithm on the observed data with limits. Additionally, the effect of unsupervised characteristics on clustering was recognised using a fundamental binning method. As a result, while the limited clustering technique improves the CA, italso raises the temporal complexity.

RESEARCH METHODOLOGY

In the case of high-dimensional data, clustering is the process of grouping data with anywhere from a few dozen to a lot of dimensions. As the number of dimensions in a dataset grows, the concept of distance becomes less precise because the distance between any two points in a given dataset converges. This is known as the curse of dimensionality. The distinction between the closest and farthest point in particular is meaningless at this point. This paper is about how well hierarchical clustering works for big datasets. In experiments, NBC works well with fake data sets that have Gaussian distributions. Algorithm 1 is the name of the new NBC clustering algorithm. From Algorithm 1, theNBC algorithm has five main steps in each run.

In the first step, a quad-tree is used to divide up the input dataset (i.e., line 3 in Algorithm 1). The second step is grouping the data with the NNB method (i.e., line 5 in Algorithm 1).The third step is to look for mutual nearest neighbor pairs and nearest neighbor pairs in all of the data groups that you have (i.e., line 6 in Algorithm 1). The fourth step is to find all the GMN pairs (i.e., line 8 in Algorithm 1). This is the last step. It's about putting together pairs that came from the fourth step (i.e, line 9 in Algorithm 1).

Algorithm 1 NBC Algorithm

1: Initial original clusters
2: **repeat**
3: Store clusters in a quad-tree
4: **for** each leaf node of the quad-tree **do**
5: Group the points in the NNB of the leaf node
6: Get mutual nearest neighbor pairs and nearest neighbor pairs of the regional points
7: **end for**
8: Globalize all mutual nearest neighbor pairs
9: Merge clusters by the pairs to new clusters
10: **until** No new clusters produced

In the NBC algorithm, the task of grouping the points in NNB of each leaf node is separatefrom the task of grouping them together. After grouping data with NNB, the computation of each group is not connected to the computation of the group next to it. So, the NBC algorithm

can be sped up by using parallel and disturbed computing frameworks to do more work at once. In Algorithm 4, you can see how the NBC parallel and distributed computing (NBCP) algorithm works.

Algorithm 2 NNB pseudo code

1: Get a region $R1$'s scope $\{(x1, y1), (x2, y2)\}$
2: Let point $A(x1, y1)$ denote $R1$'s left-bottom
3: Let point $B(x2, y2)$ denote $R1$'s right-top
4: dAB denotes distance between A and B
5: $xb1 = x1 - d_{AB}$
6: $yb1 = y1 - d_{AB}$
7: $xb2 = x2 + d_{AB}$
8: $yb2 = y2 + d_{AB}$
9: return $R1$'s NNB scope $\{(xb1, yb1), (xb2, yb2)\}$

Multiple processes are used to group each leaf node in the quad-tree into groups, and then the MapReduce model is used to find the mutual nearest neighbor pairs for each group (i.e,line 6 in Algorithm 4). People who use parallel and distributed computing can improve the performance of NBC.There is a very high number, based on NNR, that a point and its closest neighbor will be classified as the same thing when there are many data points.

Algorithm 4 The NBCP algorithm

1: Initial original clusters
2: repeat
3: Store clusters in a quad-tree
4: Use multi-process to group the points in the NNB of the leaf node
5: Get grouped data files
6: Use Map and Reduce model to calculate mutual nearest neighbour pairs of the grouped points
7: Globalize all mutual nearest neighbour pairs
8: Merge clusters by the pairs to new clusters
9: until No new clusters produced

New NNB pseudo-code

1: for each point in region do
2: Get each point's NNB scope $\{(x1, y1), (x2, y2)\}$
3: if $xb1 > x1$ then
4: $xb1 \leftarrow x1$
5: end if
6: if $yb1 > y1$ then
7: $yb1 \leftarrow y1$
8: end if
9: if $xb2 < x2$ then
10: $xb2 \leftarrow x2$
11: end if
12: if $yb2 < y2$ then
13: $yb2 \leftarrow y2$
14: end if
15: end for
16: return $R1$'s NNB scope $\{(xb1, yb1), (xb2, yb2)\}$

NNM may be utilized to accelerate computation. The NNM algorithm, presented in Algorithm 5, merges NN pairings into subclusters. When NNM is utilized to cluster original clusters to sub clusters in then NBC algorithm, the new algorithm is termed NBC fast (NBCF) algorithm.

Algorithm 5 NNM algorithm pseudo-code

1: for each NN pair (*P1*, *P2*) in NN pairs do
2: if (*P1* ∈ *cluster A* and *P2* /∈ any cluster) then
3: Add *P2* into *cluster A*
4: end if
5: if (*P2* ∈ *cluster B* and *P1* /∈ any cluster) then
6: Add *P1* into *cluster B*
7: end if
8: if (*P1* and *P2* /∈ any cluster) then
9: Create new cluster *C*
10: Add *P1* and *P2* into *cluster C*
11: end if
12: end for

Result and Discussion

We show the results of an experimental data of our algorithmic proposals on both syntheticand real-world data sets. Our trials are done on a server with 2.5 GHz 24 cores Intel Xeon CPU E5-2640. We employ 60 Hadoop computing nodes to search for mutual closest neighbor pairings in the NBCP algorithm, which runs 10 processes for grouping data by NNB.

In experiments, the outcomes of NNC, NBC, AHC and MLHC are explored and Ward's approach is employed as the connection function between two clusters. On datasets of varying sizes, the time costs of these techniques are compared. Finally, NBC and NBCP's outcomes and performances are examined.

Figure 2 shows the synthetic dataset's clustering results for the NBC, NNC, MLHC, and AHC algorithms. Figure 2 indicates that the linking function employed in NBC meets the cluster aggregate inequality, hence any MN pairings must be kept and will merge ultimately in normal AHC.

NBC and MLHC cluster the dataset in the same order by locating all MN pairings and merging them to new clusters together, then they continue this procedure until only one cluster is left, as illustrated in Figure. 2(a) and 2(c) (c). As illustrated in Figure 2, NNC searches for a pair of MN and merges them into a new cluster all at once. This procedure isrepeated until only one cluster remains (b). AHC selects the smallest distance MN pair in the dataset and merge them to a new cluster at once, then repeats this procedure as illustrated in Figure. 2(d).

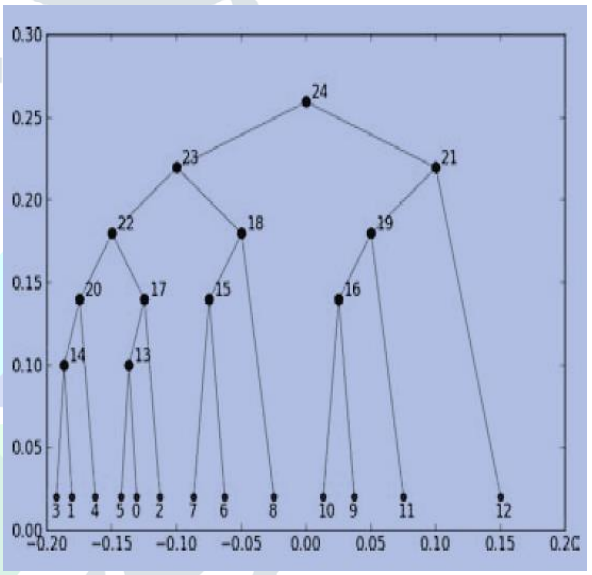

Figure 2(a) NBC dendrogram

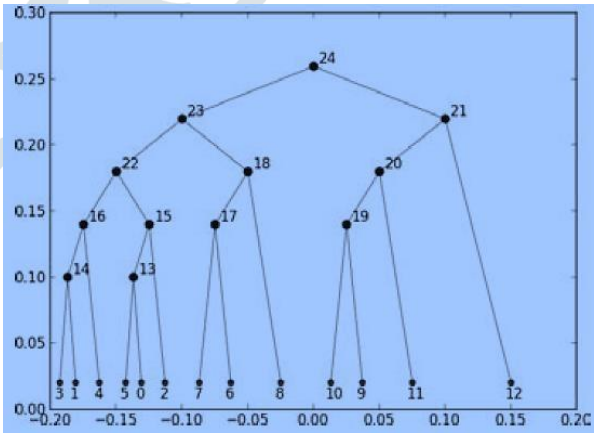

Figure 2(b) NNC dendrogram

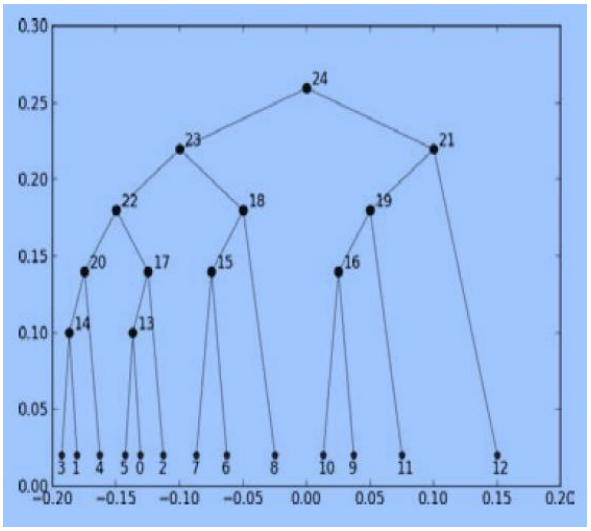

Figure 2(C) MLHC dendrogram

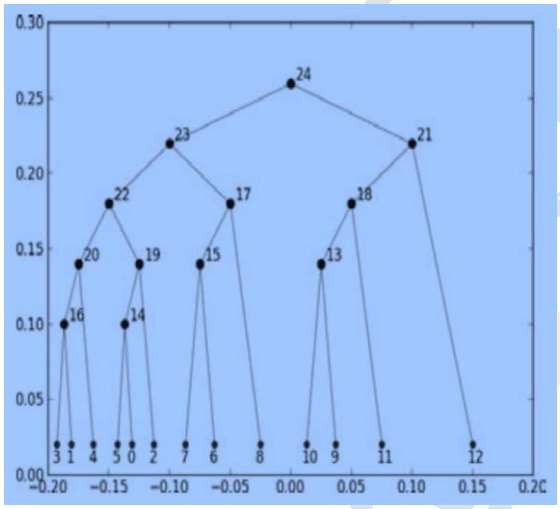

Figure 2(d) AHC dendrogram

To conclude, NBC computes precisely the same clustering results as those of the MLHC, NNC and AHC and algorithms, however in a different sequence, as illustrated in Figure. 2.

It is shown in Figure 3 that the NBC algorithm is better than NNC, MLHC, and AHC when it comes to how well it works. Figure 3 shows that NBC is much faster than the other algorithms. When the number of synthetic data items grows to more than 9000, the time costs of NNC, MLHC, and AHC are not worth the time they will take.

NBC and NBCP performance comparison

Shown in Figure 3, the time costs of NBC are more than those of NBCP. This happens when there are lots of "synthetic" data items (more than 180k). This makes NBCP number faster. In other words, when the number of data items is less than 180k, the NBC method is more efficient than the method called NBCP. These files must be moved from disk to HDFS using Hadoop. This is because data is written to disk in files. MapReduce's I/O and start-up process take a certain amount of time.

From the experiments, it looks like NBC can cut the time it takes to do a lot of work. Based on the idea of NNB, parallel and distributed computing frameworks can be used to make NBCP run faster. Our new algorithms make clustering large datasets a lot more practical, so it can be done.

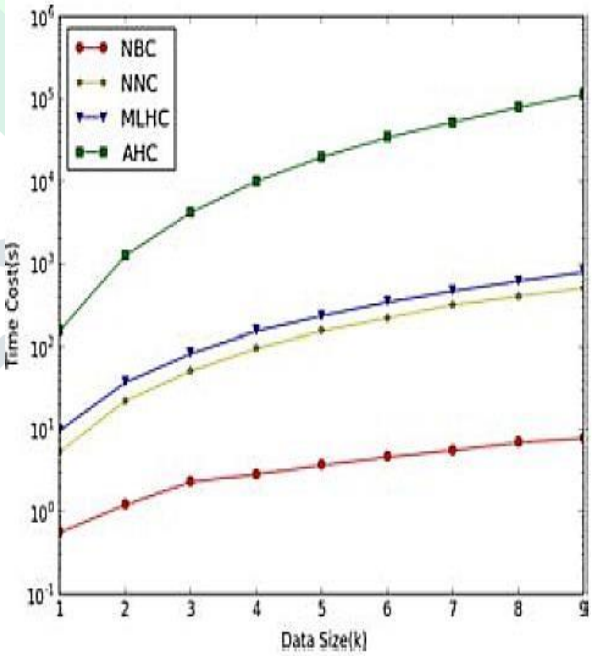

Figure 3(a)

Data Items	NBC	NNC	MLHC	AHC
1k	0.5s	5.4s	9.8	155.6s
2k	1.2s	22.6s	37.7	1263.2s
3k	2.3s	51.7s	84.3	4234.1s
4k	2.8s	97.3s	154.4	10133.4s
5k	3.7s	154.3s	235.4	19884.6s
6k	4.6s	221.8s	344.5	34405.2s
7k	5.6s	319.8s	469.5	52888.1s
8k	7.0s	406.1s	618.3	79824.8s
9k	7.9s	509.2s	796.4	116178.1s

3(b) NBC,NNC,MLHCand AHC performance comparison

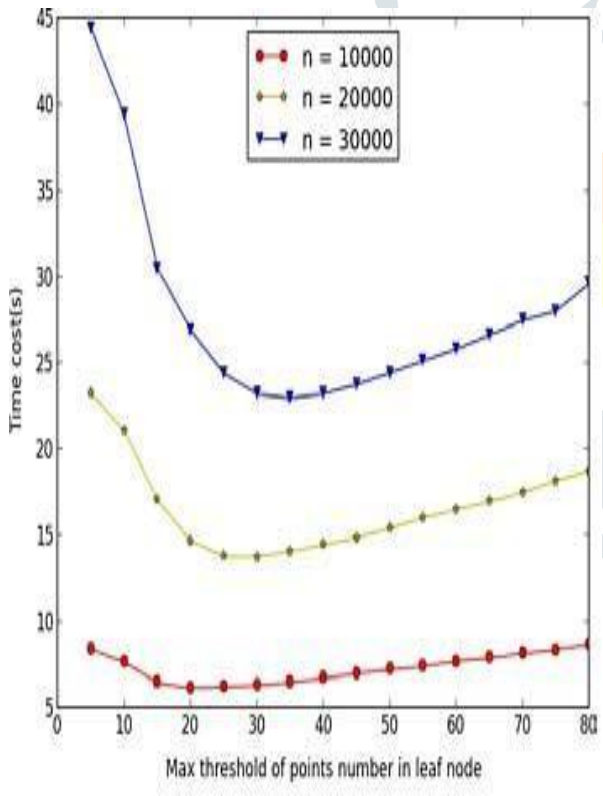

Figure 4(a)

Data Size	m^*	m	T_{max}	Time cost
10k	3.5	9.7	20	6.1s
20k	3.6	18.5	30	13.7s
30k	3.7	19.7	35	22.9s
40k	3.7	25.2	45	31.8s
50k	3.7	16.5	45	42.9s
60k	3.8	15.4	45	53.7s
70k	3.8	17.0	35	64.4s
80k	3.8	19.5	45	74.6s
90k	3.8	21.9	45	86.1s

Figure 4(b)

4(a) NBC performance time cost in various data sizes n and average number of leaf nodes m, and (b) Optimal solutions of m compared to experimental results of the NBC sequential algorithm in various data sizes.

The experimental findings of parameter assessment are equivalent to the algorithm analysis in the NBC sequential algorithm. However, due to the complexity of the NBC parallel algorithm and the parallel and distributed computing infrastructure, the practical results of parameter assessment deviate the algorithm analysis. The time costs associated with NBCP gradually grow as data size increases.

Conclusion

Clustering huge datasets is a pervasive problem. This paper discusses partitioning huge datasets and presents an effective hierarchical clustering technique called NBC. By using NNB, NBC may significantly minimize the time complexity of hierarchical. NBC excels in two-dimensional space. Because each set of data in NNB is distinct from the others, multi-process and MapReduce models may be utilized to speed NBCP processing. Nearest neighbour search is a critical technique in a wide variety of industries. The notion of NNB enables an effective approach of searching for closest neighbours in huge datasets. In future study, we may use NNB to handle a variety of additional issues involving enormous datasets.

REFRENCES:

1. B. Bozdemir, S. Canard, O. Ermis, H. Möllering, M. Önen, and T. Schneider, "Privacy-preserving density-based clustering," 2021.View at: Google Scholar

2. L. Wang, H. Wang, W. Zhou, and X. Han, "A novel adaptive density-based spatial clustering of application with noise based on bird swarm optimization algorithm," Computer Communications, vol. 6, 2021.View at: Google Scholar

3. Karwan Qader, Mo Adda and Mouhammd Al-kasassbeh (2017), "ComparativeAnalysis of Clustering Algorithms in Network Traffic Faults Classification", International Journal of Innovative Research in Computer and CommunicationEngineering, Vol. 5, No.4, pp.6551-6563.

4. Yu Wang, Yang Xiang, Jun Zhang, Wanlei Zhou and BailinXie(2016), "Internet traffic clustering with side information", Journal of Computer and System Sciences, Elsevier, Vol. 80, No.5, pp.1021–1036.

5. M. J. Reddy and B. Kavitha, "Clustering the mixed numerical and categorical dataset using similarity weight and filter method," *International Journal of Database Theory and Application*, vol. 5, pp. 121–134, 2012.

6. Y. Wei, X. Zhang, Y. Shi et al., "A review of data-driven approaches for prediction and classification of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1027–1047, 2018.

7. M. Kumar, P. Chhabra, and N. K. Garg, "An efficient content based image retrieval system using BayesNet and *K*-NN," *Multimedia Tools and Applications*, vol. 77, no. 16, pp. 21557–21570, 2018.

8. A. Onan, S. Korukoğlu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," *Information Processing & Management*, vol. 53, no. 4, pp. 814– 833, 2017.

9. Luiz Fernando Carvalhoa, Sylvio Barbona, Leonardo de Souza Mendes and Mario Lemes Proença (2016), "Unsupervised learning clustering and self-organized agents applied to help network management", Expert Systems with Applications, Elsevier, Vol. 54, pp.29-47.

10. P. Dahiya and D. K. Srivastava, "A comparative evolution of unsupervised techniques for effective network intrusion

detection in hadoop," in *Proceedings of the International Conference on Advances in Computing and Data Sciences*, pp. 279–287, Dehradun, India, April 2018.

11. Mohiuddin Ahmed and Abdun Naser Mahmood (2015), "Novel Approach forNetwork Traffic Pattern Analysis using Clusteringbased Collective Anomaly Detection", Annals of Data Science, Springer, Vol. 2, No.1, pp.111–130.

12. S. Mehrotra and S. Kohli, "Comparative analysis of *K*-means with other clustering algorithms to improve search result," in *Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 309–313, Delhi, India, October 2015.

13. Shital Salve and Sanchika Bajpai (2014), "Online stream mining approach for clustering network traffic", IJRET: International Journal of Research in Engineering and Technology, Vol. 3 No.2, pp.300- 304.X. Zheng and N. Liu, "Color recognition of clothes based on *K*-means and mean shift," in *Proceedings of the Intelligent Control, Automatic Detection and High- End Equipment (ICADE)*, pp. 49–53, Beijing, China, July 2012.

14. Lin Guan-zhou, XIN Yang, NIU Xin-xin and JIANG Hui-bai (2010), "Network traffic classification based on semi-supervised clustering", The Journal of China Universities of Posts and Telecommunications, Elsevier, Vol. 17, Supplement 2, pp.84-88.

15. Lin Guan-zhou, XIN Yang, NIU Xin-xin and JIANG Hui-bai (2010), "Network traffic classification based on semi-supervised clustering", The Journal of China Universities of Posts and Telecommunications, Elsevier, Vol. 17, Supplement 2, pp.84-88.