

Comparative Weka Analysis of Clustering Algorithm's

Harjot Kaur

CT Group of Institution/CSE, Jalandhar, 144041, India
E-mail: Harjotkaur844@gmail.com

Er. Prince Verma

CT Group of Institution/CSE, Jalandhar, 144041, India
E-mail: prince.researchwork@gmail.com

Received: 29 March 2017; Accepted: 19 April 2017; Published: 08 August 2017

Abstract—Data mining is a procedure of mining or obtaining a pertinent volume of data or information making the data available for understanding and processing. Data analysis is a common method across various areas like computer science, biology, telecommunication industry and retail industry. Data mining encompass various algorithms viz. association rule mining, classification algorithm, clustering algorithms. This survey concentrates on clustering algorithms and their comparison using WEKA tool. Clustering is the splitting of a large dataset into clusters or groups following two criteria ie. High intra-class similarity and low inter-class similarity. Every cluster or group must contain one data item and every data item must be in one cluster. Clustering is an unsupervised technique that is fairly applicable on large datasets with a large number of attributes. It is a data modelling technique that gives a concise view of data. This survey tends to explain all the clustering algorithms and their variant analysis using WEKA tool on various datasets.

Index Terms—Data Mining, Clustering, Partitioning Algorithm, Hierarchical Clustering Algorithm, CURE, CHAMELEON, BIRCH, Density Based Clustering Algorithm, DENCLUE, OPTICS, WEKA Tool.

I. INTRODUCTION

Recently, the generation and collection of data for various purposes had increased rapidly. Commercial sites, business sites, government sites, bank transaction, scientific and engineering field, social media like Facebook, Twitter have provided us with a large amount of data. This increase in the size and complexity of the data leads to the generation of various new tools and techniques so as to process the data and to extract hidden information and knowledge from the data. This leads to the term called as Knowledge Discovery in Databases (KDD) [3], [4].

Data mining, which is also referred to as knowledge discovery in databases, means a process of extraction of unknown and potentially useful information (such as

rules of knowledge, constraints, regulations) from data in databases [5]. Data mining is simply a process of discovering unknown hidden information from the already known data.

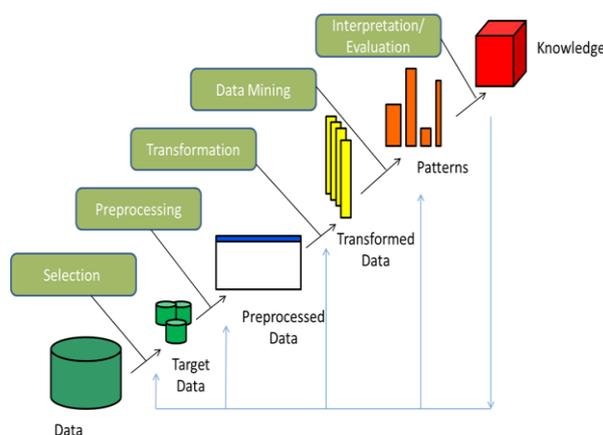


Fig.1. Knowledge Discovery Process in Data Mining [1].

As depicted in Fig. 1, the first step in data mining for discovering hidden patterns and extracting the knowledge is called as Extraction. Once the data is extracted the next step is to clean the data for processing. Cleansing of data is carried out to remove noise and undesirable feature. The process is called as ECTL (Extracting, Cleaning, Transforming and Loading). Handling static data is easier than dynamic data. The data nowadays have various forms like structured data (text data), unstructured data (audio, video, images). To transform this data to desirable properties different techniques are carried out like classification, clustering. The paper will discuss some of the data mining clustering techniques [12], [13].

A. Data Mining Algorithms

Data Mining follows three main steps in preparing the data for processing, reducing the data to concise it and extracting useful information. The major algorithms followed in data mining are classified into six classes [1]. The following steps are executed on raw data to obtain relevant information.

1. *Anomaly Detection*: It is the identification of data records that are not desirable and might contain an error in it, say temperature is 45, this indicates a bogus data without units.
2. *Association Rule Mining (ARM)*: It is a process of identifying linkage among the items present in the database. ARM induces the relationship between the items, say bread and butter or bread and jam.
3. *Clustering*: A descriptive process that groups the data of same structure in one cluster without using a pre-defined structure say, a mail is a spam or ham mail. Clustering will group the set of data into two clusters based on the characteristics generated viz. a mail can be spam depending upon the type of content in the mail or a mail can be ham mail. Such as K-Means and K-Medoid.
4. *Classification*: A predictive process that generalizes the known structure to new data. Such as Support vector machine, Multi-Layer Perceptron.
5. *Summarization*: A process of representing the data in the compact form for visualization.

B. Data Mining Techniques

Various data mining techniques and systems are available to mine the data depending upon the knowledge to be acquired, depending upon the techniques and depending upon the databases [1].

1. *Based on techniques*: Data mining techniques comprises of query-driven mining, knowledge mining, data-driven mining, statistical mining, pattern based mining, text mining and interactive data mining.
2. *Based on the database*: Several databases are available that are used for mining the useful patterns, such as a spatial database, multimedia database, relational database, transactional database, and web database.
3. *Based on knowledge*: Fig. 1 depicts the knowledge discovery process, including association rule mining, classification, clustering, and regression. Knowledge can be grouped into multilevel knowledge, primitive knowledge and general level knowledge.

II. CLUSTER ANALYSIS

The simple means for analysing and managing large volume and complexity of data is to classify or group the data based on predefined categories or unlabelled clusters items. "Generally classification technique is either supervised or either it is unsupervised technique solely depending on whether they assign new items to one of a finite number of discrete supervised classes or unsupervised categories respectively [2], [7], [9], [10], while clustering is an unsupervised approach. The aim of clustering is to partition a finite unlabelled data set into a finite and discrete set of data structures that are hidden, rather than providing exact properties of unobserved

samples generated from the same probability of distribution [2], [8], [9]. Clustering mainly follows two criteria for partitioning the unlabelled data items that include high similarity between the data objects within the same clusters and low similarity outside the clusters. Clustering leads to the formation of hard clusters.

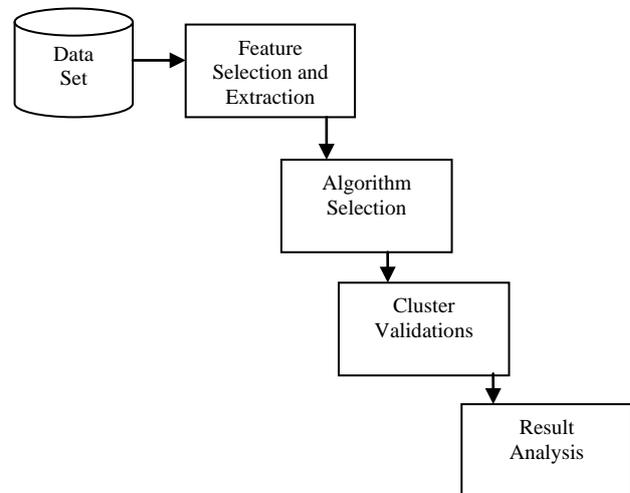


Fig.2. Steps for the formation of clusters and representing data inside clusters [2], [11].

It is important to learn the difference between classification and clustering. Both these terms can be explained by a simple example. Let a bucket contain some fruits say orange, grapes, apple and banana. Classification works on predefined knowledge or set of information. Classification algorithm will choose any feature like colour to be one of those features and will categorize the fruits depending on that set of information, while clustering has no such model for grouping the objects. Clustering defines its own model say shape to be one of those models. Clustering will group the above fruits based on shape. Clustering process can be carried out [2], [11].

1. *Feature Selection or Extraction*: As referred in [2], [11] feature selection is the selection of relevant features from irrelevant features while feature extraction is the production of new features from predefined features.
2. *Selection and Designing of Clustering Algorithm*: No such clustering algorithm is available which solves all the clustering problems. Different algorithms are available with different distance formula like Euclidean distance, Manhattan distance, Minkowski distance. It is important to first understand the problem carefully and then decide which algorithm is best suitable.
3. *Cluster Validation*: Cluster validation is the assessment of the clusters formed. Testing of the clusters is done to make sure about the quality of the clusters so formed and guarantees that desirable clusters are achieved. Testing of the clusters can be done by three ways external indices, internal indices and relative indices [2], [6]

depending upon the type of clusters.

4. *Result Analysis:* The results so formed from the original set of the data is analysed to have an insight view of it and to ensure qualities of clusters formed are satisfied.

III. PARTITIONING CLUSTERING ALGORITHM'S

Different initial points in clustering lead to different results [2], [14], [15]. Depending upon the properties of clusters formed clustering algorithms can be classified as Partitioning Algorithms, Hierarchical Algorithms and Density Based Algorithms [2], [15], [18].

A. Partitioning Algorithm

K-Means is the most famous and common type of clustering algorithm used to assign data objects to a cluster. Partitioning algorithm works by dividing the data objects into 'k' number of clusters partitions. Let dataset 'D', contains 'n' number of data items where 'k' is the number of partitions, partitioning algorithm assign 'n' number of data items to 'k' partitioners where ($k \leq n$). These 'k' partitions are called as Clusters. The data items in a single cluster possess similar characteristics. The number of clusters formed should not be similar to each other as depicted in Fig. 3. Partitioning algorithms ensure that no cluster should be empty [17]. Partition algorithm does not follow any hierarchy like hierarchical algorithm follows, it divides the data objects in a single level. Partitioning works efficiently on a large dataset. A common criterion function for generating clustering output in partitioning algorithm is by using squared error algorithm [19].

$$E^2 = \sum_{j=1}^k \sum_{i=1}^n (\|x_i^{(j)} - c_j\|)^2 \quad (1)$$

Where $(\|x_i^{(j)} - c_j\|)^2$ is a chosen Euclidean distance, while 'c_j' are the cluster centres, 'x_i^(j)' represent the data objects.

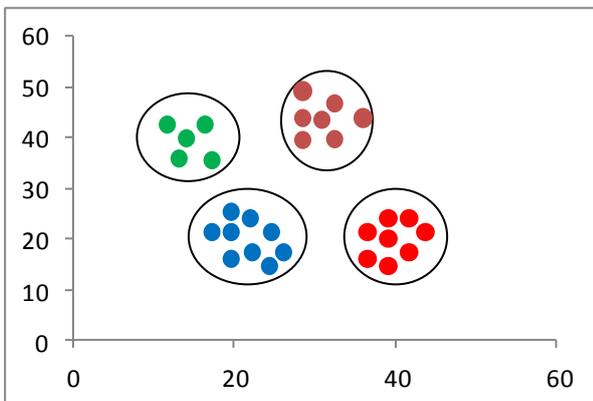


Fig.3. Objects partition in different clusters, where k=4 representing different data in the different cluster.

B. K-Means Clustering Algorithm

K-Means algorithm is the simplest algorithm which works on iterations to group the data objects in clusters [2]. Following the criteria of the sum of square error, K-Means algorithm has a time complexity of $O(n)$. Trapping in local optima is the biggest problem with K-Means algorithm if the initial clusters are not chosen with care. K-Means algorithm aims at minimising the squared error. It is an unsupervised learning in which the 'k' is fixed apriori defines the initial cluster centre for every cluster the centroid should be defined careful and cunningly because the change in location also leads to change in results. The best way to place the cluster centres is to place them far from each other [16]. After the assignment of centres to the initial clusters, the next step is to associate each nearest points to a particular cluster. When all the data points are placed in the cluster recalculate a new centroid for the clusters. Repeat the process again until no data object changes its location. K-Means algorithm converges faster.

C. Algorithmic Steps for K-Means Algorithm

Let $D = (d_1, d_2, d_3, \dots, d_n)$ be the data points in a dataset and $S = (s_1, s_2, s_3, \dots, s_n)$ be the group of centres. The algorithm follows the below-mentioned steps for clustering [16]:

1. Manually and randomly select the initial cluster centres 'c'.
2. Use Euclidean distance criterion function and calculate the distance between all the data points and the clusters centres.
3. Information points with minimum distance from cluster centres are placed in that particular cluster.
4. Above process is carried out for all the data points until all the points are placed in the clusters.
5. Recalculate the new cluster focus 'c_i' using equation mentioned below and again calculate the distance of every information point from new cluster centres.

$$S_i = (1/c_i) \sum_{j=1}^{c_i} d_i \quad (2)$$

Where, 'c_i' is the number of data points in the ith cluster

6. If no new reassignment of data points takes place then stop the process, if yes repeat the steps 2 and 3.

D. Advantages of K-Means Algorithm

1. K-Means is simple to implement and robust algorithm.
2. K-Means Algorithm converges faster than any other algorithm.
3. The algorithm can work efficiently on large datasets and with a large number of attributes.
4. Tighter and arbitrarily shapes of clusters are formed by K-Means clustering algorithm.
5. Compute faster than Hierarchical clustering algorithm.

E. Disadvantages of K-Means Algorithm

1. The algorithm cannot work on categorical data.
2. Trapping into local optima is the biggest disadvantage of K-Means algorithm.
3. Selection of initial clusters and its centroids are done manually and randomly which leads to variation in results if not done carefully.
4. The Non-linear dataset cannot cooperate with K-Means algorithm.
5. Numbers of initial clusters are fixed apriori.
6. Noise and outliers affect K-Means algorithm.

F. K-Medoid Algorithm

K-Medoid algorithm is an extension of K-Means algorithm. Cluster representation in K-Medoid is carried out by one of its objects. As peripheral information points/objects have no effect on Medoids so they are insensitive to outliers and support all types of attributes [14]. The basic idea behind the K-Medoid algorithm is to firstly find the Medoids by random means from 'n' objects for 'k' number of clusters. Medoids in the initial clusters are most centrally located data objects. The leftover information points are placed in similar clusters. K-Medoid avoids taking the mean value of information points labelled inside the clusters instead it uses representative points as reference points [18]. K-Medoid focuses on minimizing the sum of dissimilarity between the information objects labelled in the clusters.

G. Algorithmic Steps for K-Medoid Algorithm

Let dataset $D = (d_1, d_2, d_3, \dots, d_n)$ contains 'n' objects that are to be partitioned into 'k' number of clusters. K-Medoid algorithm minimizes the sum of dissimilarities [16].

1. Initialization: Randomly choose 'k' clusters of 'n' objects from 'D' dataset as the initial Medoids.
2. Assignment Step: Assign each similar object to the cluster with the minimum distance to its Medoid 'M'.
3. Arbitrarily select a non-Medoid object 'O'.
4. 'S' Swap the Medoid 'M' and 'O' to compute the total cost of configuration which is the average dissimilarity of 'O' with all the data points labelled to 'M'.
5. Select the Medoid of the lowest cost of configuration.
6. If $S < 0$ then swap 'M' with 'O' to form a new set of Medoid.
7. Repeat until no change.

H. Advantages of K-Medoid Algorithm

1. K-Medoid is more robust to noise and outliers as compared to K-Means algorithm.
2. K-Medoid minimizes the sum of dissimilarity instead of the sum of squared error which makes it faster than the K-Means algorithm due to the decrease in the number of comparisons.

I. Disadvantages of K-Medoid Algorithm

1. Initial clustering problem and selection of objects still prevails.
2. As compared to K-Means algorithm, K-Medoid algorithm is not suitable for large datasets.
3. The cost of the K-Medoid algorithm is high.

IV. HIERARCHICAL CLUSTERING ALGORITHM'S

Instead of having a single partition like that in Partitioning algorithm, hierarchical algorithm forms a hierarchical structure of clusters called as a dendrogram. The tree like dendrogram can be split at different levels to produce different clustering of data. Hierarchical algorithm varies from Agglomerative clustering to Divisive clustering. In Agglomerative clustering algorithm, bottom-up approach is followed. This type of clustering assumes every document to be single clusters so as to group all the pairs of clusters in a single cluster that contains all the documents. Top down clustering proceeds by splitting the clusters recursively from a single cluster to multiple clusters. This approach is termed as divisive clustering algorithm [18].

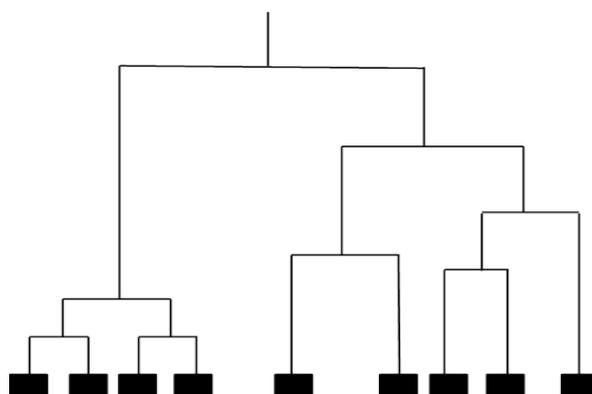


Fig.4. Representation of Hierarchical clustering using Dendrogram.

A. Algorithmic Steps for Hierarchical Agglomerative Clustering

Given a set of 'I' data objects to be clustered and $I \times I$ matrix of distance based on similarity. S.C. Johnson in 1967 defined the hierarchical algorithm [27]:

1. Initialization: Assign each object to a cluster in such a way that if you have 'I' objects then the number of clusters will be 'I'. Each cluster is labelled by one object and the distance measures between the clusters are directly proportional to the distance between the objects they contain.
2. Select a pair of clusters that have minimum distance (most similar) between them using single linkage metrics (similarity of most similar points) or complete linkage metrics (similarity of most dissimilar objects) and merge them in a single cluster. At this point, the numbers of clusters left are (I-1).

3. Based on the distance (similarities), compute for each new cluster formed and previously formed clusters.
4. Until all the clusters are not labelled in a single cluster of size 'I' repeat step 2 and 3. Once all data objects are placed in a single cluster of size 'I'.

B. Algorithmic Steps for Divisive Clustering

Divisive clustering is further categorized into monothetic and polythetic. Monothetic clustering selects one attribute at a time to split the cluster while polythetic clustering employs all the attributes in a set of data to split the clusters. The Polythetic clustering algorithm is as defined below [20].

1. Assign all the objects in a single cluster to reduce the clusters to a singleton cluster.
2. Measure the distance between the objects based on any criterion and create corresponding distance matrix. The distance should be sorted in an ascending order. Also, choose a threshold distance.
3. Select two objects that have the maximum distance between them that are most dissimilar to each other.
4. If the distance between the two objects is less than the predefined threshold and no more splitting is possible then, stop the algorithm.
5. Create new cluster from previous clusters.
6. If a cluster is left with only one object stop the algorithm else repeat step 2.

C. CURE

Clustering Using Representative (CURE) is an Agglomerative clustering based method that works efficiently on large datasets and is robust to noise and outliers. CURE can easily identify the clusters of any shape and size. CURE intakes both partitioning clustering and hierarchical clustering which increases the scalability issues by sampling the data and partitioning the data. CURE is capable of creating 'p' partitions which enable fine clusters to be partitioned first. Secondly, CURE represents clusters by scattered points fixed apriori instead of a single centroid. CURE intakes numerical attributes. The consumption of memory for the selection of initial clusters is low in case of CURE [14], [20]. The algorithmic steps for CURE are [20].

1. Collect a sample of the dataset.
2. Split the sample into 'p' partitions of size 'n' to increase the speed of the algorithm by first performing clustering on each partition made.
3. Label each data point to the partitions by using hierarchical clustering algorithm.
4. Outliers are removed by first removing those clusters that are very slow in growing and later by removing clusters that are small in size.
5. At the end label all the data to the sample. The processing is carried out in main memory so make sure to input representative point of the clusters only.

6. Stop the algorithm after convergence is achieved.

D. CHAMELEON

An Agglomerative approach of hierarchical clustering makes use of linked graph corresponding to K-nearest neighbour. A graph is created in which each data points are linked to its K-nearest neighbour. Partition of the graph is carried out recursively to split it into small unconnected sub graphs. Each sub graph is treated as an individual sub-cluster and an agglomerative approach merges two similar clusters into one. To merge two sub-clusters interconnectivity and closeness of the clusters are taken into considerations. CHAMELEON is more efficient than CURE in finding arbitrary shapes of clusters [20], [21].

1. Consider a data set $D = (d_1, d_2, d_3, \dots, d_n)$ and 'd1 to dn' be the data points in a dataset.
2. The dataset is to be partitioned to form clusters.
3. Partition the dataset to form initial clusters using any graph partitioning approach.
4. Merge the clusters to grow a big cluster by agglomerative technique taking into account inter-connectivity and closeness.
5. CHAMELEON is a feature driven approach.

E. BIRCH

Balanced Iterative Reducing Clustering Hierarchies (BIRCH) uses CF-tree data structure to compress the data into small clusters. BIRCH is an unsupervised algorithm that works on the large set of numerical data. BIRCH is highly scalable and is capable of undoing all the changes made during initial steps. Sub-clusters are represented by cluster- features (CF) and are stored in leaves. CF tree is a tree for balancing the height that constitutes two parameters threshold for the radius and balancing factor for the leaf nodes. A single scan of the dataset is required to build a CF tree. BIRCH is fast and efficient but lacks in handling the clusters of varying shape and size as it uses diameter to handle the boundaries of clusters [21].

1. Construct a CF tree to load into memory.
2. Set desirable range by constructing small CF tree.
3. Sub-clusters to be re-clustered by using global clustering.
4. Take the centroids of the clusters formed from phase 3 as seed clusters.
5. Apply any existing algorithms to link data objects to its nearest seed to form a cluster.
6. Stop algorithm.

F. Advantages of Hierarchical Clustering Algorithm

1. The Hierarchical clustering algorithm is easy to implement.
2. Multi- point partition of data objects takes place
3. No pre-knowledge about the number of clusters is required.
4. The Hierarchical algorithm is flexible regarding the level of granularity.

G. Disadvantages of Hierarchical Clustering Algorithm

1. Difficult to handle noise and outliers
2. Once the changes regarding merging or splitting of clusters are made then the algorithm is unable to undo those changes.
3. In some case, dendrogram makes it difficult to correctly identify the number of clusters.
4. Termination criterion of the algorithm is not definite.
5. The algorithm turns out to be expensive in case of large datasets.

V. DENSITY BASED CLUSTERING ALGORITHM'S

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) proposed by Martin Ester is a density based clustering algorithm which is capable of removing noise from spatial datasets and is efficient in finding arbitrary shapes of clusters [22]. It grows clusters depending upon the density of the objects present in its neighbourhood [23]. The implementation of partitioning the data objects are carried out by concepts like connectivity, density-reachable [14]. A point 'p' is said to be density-reachable from point 'q' is within 'e' distance of point 'q' and point 'q' also contain enough number of points within 'e' distance while 'p' and 'q' are considered as density connectivity if both 'p' and 'q' are within the 'e' distance and there exist a point 'r' which have enough number of points in its neighbour. A chain is formed in which each point corresponds to each other [24]. Both these concept depends upon epsilon neighbourhood 'e' that control the size of clusters and its neighbourhood and also on minpts.

A. Algorithmic Steps for Density-Based Clustering

Let $D = (d_1, d_2, d_3, \dots, d_n)$ be the data points in a dataset. DBSCAN algorithm undertakes two parameters 'e' and minpts (minimum number of points corresponding to a cluster) to form a cluster [24].

1. Initially, start from any random point that is not visited.
2. Explore the neighbours of undertaken initial point by using 'e'. The points nearby 'e' distance are the corresponding neighbours.
3. If the initially random selected point has enough neighbouring points, then clustering process starts and all its corresponding neighbouring points are marked as visited otherwise the point will be considered as noise. Later this point can found in a cluster.
4. If a point seems to be a part of the cluster, then its 'e' neighbourhood points are also the part of the cluster. Step 2 is repeated until all the points neighbouring to the initial point is placed in one cluster.
5. A new point that is not visited is identified and the same process is repeated for the formation of further clusters or noise. Stop when all the points

are visited.

B. DENCLUE

Clustering Based on Density Distribution Function (DENCLUE) is a clustering technique that is completely dependent on density function. The two ideas clinched by DENCLUE is that each data object's influence can be described by using an influence function that describes the impact of data objects with its neighbours. Secondly, the overall density of data space is the summation of influence function of all the datasets. Cluster identification in DENCLUE is carried out by using density attractors where these attractors are the local maxima of influence functions. The computation of overall density in all the datasets is carried by forming a grid structure. DENCLUE is resistance to noise and insensitive to data ordering along with the feature of forming arbitrary shapes of clusters makes it unique but on the other side, it lacks in describing the clusters description properly. Thirdly, it is not a perfect choice for high dimensional datasets [20], [21]. DENCLUE algorithms execute in two steps:

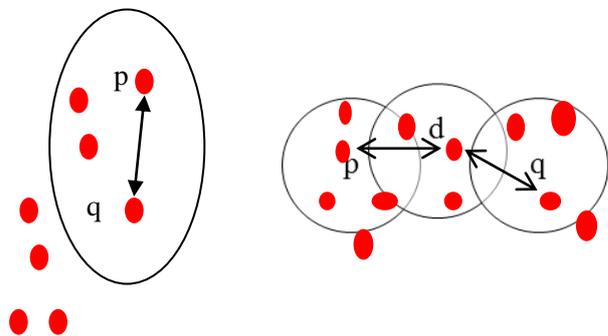


Fig.5. a) Representing Density-Connected between point's 'p' and 'q' via 'd'. b) Representing Directly Density Reachable from point 'p' to 'q'.

1. *Pre-clustering phase*: Map construction for the relevant and desirable workspace is constructed. The aim to construct the map for the relevant workspace is to boost the calculation for the density function to access the nearest neighbours.
2. *Clustering phase*: In the final clustering phase, identification if density attractors are made and the related density attracted points are identified.

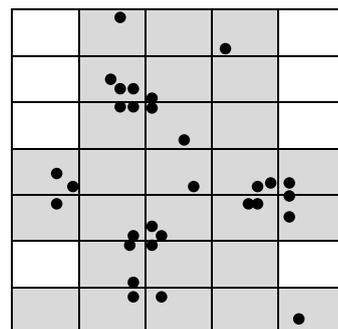


Fig.6. Map construction in a workspace representing data points.

C. OPTICS

Ordering Points to Identify the Clustering Structure (OPTICS) is another clustering technique for forming clusters by using density function in spatial datasets. OPTICS follows the same path of DBSCAN but it converts one of the major limitation of density based clustering as its advantage i.e. the problem of detecting meaningful and informative clusters from the data having varying density. In OPTICS clustering is carried out by placing the objects in such a way that objects that are close to each other become neighbours in the ordering. A dendrogram representation is structured that constitutes a special distance between each point containing the density of the points that is to be accepted by the clusters to allow both the points to belong to the same cluster [20].

D. Advantages of DBSCAN

1. DBSCAN handles noise and outliers efficiently.
2. The number of clusters is not specified in priori like that in the case of K-Means algorithm.
3. DBSCAN is capable of finding arbitrary shapes of clusters and can find clusters surrounding each other but are not connected.
4. The parameters minpts and distance threshold can be set by an individual.

E. Disadvantages of DBSCAN

1. Computation cost is high.
2. DBSCAN intakes only two parameters and is highly sensitive to the parameters.
3. Proper cluster description is not carried out by DBSCAN.
4. Not suitable for high dimensional datasets as the quality of clusters depends on distance measure and distance measure uses Euclidean distance which cannot render appropriate results with high dimensional datasets.
5. DBSCAN lacks in handling clusters with varying densities.

VI. RESULT ANALYSIS

WEKA is widely accepted in various domains like business and academia. WEKA project was aided by the New Zealand government [25]. In 1993, the University of Waikato in New Zealand began improvement of the initial WEKA (which came out as a combination of TCL/TK, C and Makefiles) [26]. Waikato Environment for Knowledge Analysis (WEKA) was envisioned to provide the researcher with a toolbox of learning algorithm along with a framework to implement new algorithms. WEKA being an open source environment is used for both machine learning and data mining. WEKA provides an analysis environment for various algorithms

like clustering, regression, classification and association rule mining. Attribute relation file format (.ARFF) or Comma separated value (CSV) format is opted by WEKA for information processing. In case the set containing information is not in the format we need to change it. WEKA allows users to interact with various graphical user interfaces like Explorer, Experimenter, Knowledge Flow and Simple CLI. The main interface is Explorer which constitutes various data mining tasks like pre-processing of the information set, classification, clustering, and association rule mining.

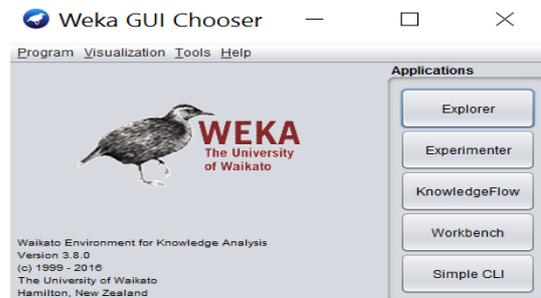


Fig.7. View of WEKA Tool.

The other main interface viz. experimenter is designed to perform an experimental comparison of the observational performance of algorithms based on many different evaluation parameters that are available in WEKA [25].

A. Results Analysis on Traffic Dataset

Weka result analysis is performed on personally collected traffic dataset containing 793 instances to be labelled in different clusters. Various clustering algorithms are implemented on traffic dataset among which K-Means and K-Medoid demonstrated better performance than other clustering algorithms. It was analysed that with the increase in the number of clusters performance of clustering algorithms degrades because of the problem of selecting the initial number of clusters and centroids. Partitioning algorithms tend to show better performance over other clustering algorithms with high accuracy performance with low execution time. The problem of failing to eliminate the overlapping of the objects degrades the performance of the hierarchical algorithms while in the case of partitioning algorithms measure are needed to be considered to solve the initial clustering problem of the algorithms to avoid the variations in the results. Graphical representation of accuracy and F-Measure comparison and time comparison in Fig. 8, 9 and 10 on traffic dataset with 793 instances demonstrated that K-Medoid algorithm is highly accurate in labelling the data to the specified clusters with low execution time.

Table 1. Comparison Between Various Algorithms for the number of clusters k=2.

Algorithm	Accuracy	F-Measure	Time Taken	Cluster Instances
K-Means	81.37 %	83.2 %	0.966 sec	0: 180(23%) 1: 613(77%)
K-Medoid	85.70 %	86.7 %	0.429 sec	0:613(77%) 1:180 (23%)
Density Based Clustering	75%	76.3%	0.12 sec	0: 180(23%) 1: 613(77%)
Hierarchical Clustering	79.23%	81.2%	0.752 sec	0: 613(77%) 1:180(23%)

Table 2. Comparison Between Various Algorithms for the number of clusters k=3.

Algorithm	Accuracy	F-Measure	Time Taken	Cluster Instances
K-Means	72.57%	87.3 %	0.767ssec	0: 33(4%) 1: 180(23%) 2: 580(73%)
K-Medoid	87.12%	96.8%	0.588 sec	0:33(4%) 1:180(23%) 2:580(73%)
Density Based Clustering	71.56%	73%	0.232 sec	0: 180(23%) 1: 517(65%) 2: 96(12%)
Hierarchical Clustering	75.89%	77.7%	0.724 sec	0:580(73%) 1:180(23%) 2:33(4%)

Table 3. Comparison Between Various Algorithms for the number of clusters k=4.

Algorithm	Accuracy	F-Measure	Time Taken	Cluster Instances
K-Means	65.7%	90.7%	0.5705 sec	0:180(23%) 1: 351(44%) 2: 96(12%) 3:166(21%)
K-Medoid	75.92%	96.8%	0.5538 sec	0:33(4%) 1:180 (23%) 2:479(60%) 3:101(3%)
Density Based Clustering	73.92%	75.76%	0.802 sec	0: 180(23%) 1: 351(44%) 2: 96(12%) 3:166(21%)
Hierarchical Clustering	63.87%	67.34%	0.685 sec	0:580(73%) 1:87(11%) 2:33(4%) 3:93(12%)

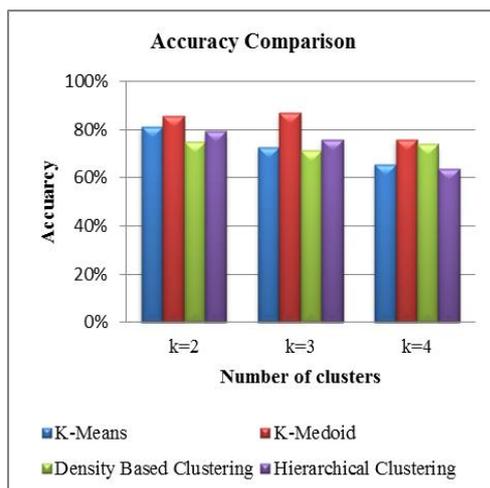


Fig.8. Accuracy comparison for traffic dataset.

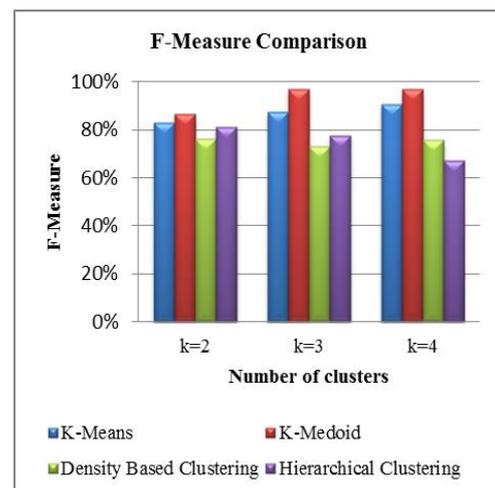


Fig.9. F-Measure comparison for traffic dataset.

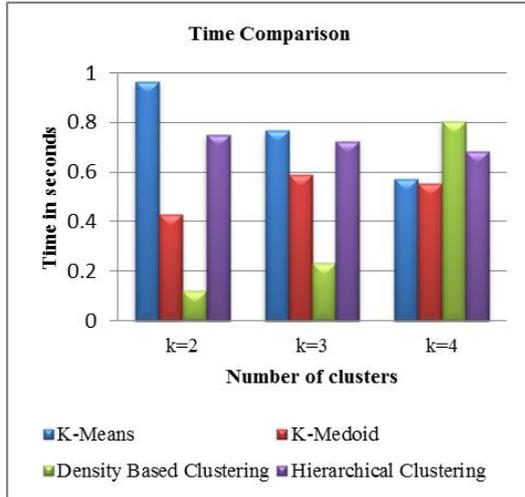


Fig.10. Time-comparison for traffic dataset.

The above-performed analysis on the traffic dataset results for the different clusters k (2, 3, and 4) demonstrated that K-Medoid represents better accuracy with low execution time. The change in the selection of the initial clusters leads to the variation in the results but still, K-Medoid can deal with noisy data, outliers and avoid the overlapping of the objects.

B. Results Analysis on Ionosphere Dataset

Weka result analysis is performed on Ionosphere dataset collected from UCI repository, containing 120 instances. K-Means and K-Medoid demonstrated better performance than other clustering algorithms. The

problem of initial selection of clusters again degrades the performance of the algorithms. Graphical representation of accuracy, F-Measure and time comparison in Fig. 11, 12 and Fig. 13 on Ionosphere dataset demonstrated that K-Medoid algorithm is highly accurate in labelling the data to the different specified clusters. The decrease in the accuracy of the K-Medoid algorithm from 76.92% for two clusters to 71.22 % for four clusters is due to the manual selection of initial clusters. The selection of the initial clusters can be dealt by using various optimization techniques like Ant colony optimization (ACO), Genetic algorithms, Artificial Bee colony optimization (ABC) or by fuzzy logics.

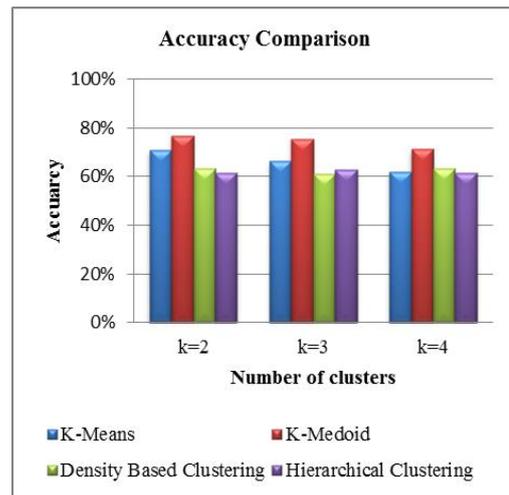


Fig.11. Accuracy comparison for Ionosphere dataset.

Table 4. Comparison between Various Algorithms for the number of clusters k=2.

Algorithms	Accuracy	F-Measure	Time Taken	Cluster Instances
K-Means	70.94%	71.5%	0.03 sec	0: 51(43%) 1: 69(57%)
K-Medoid	76.92%	77%	0.03 sec	0:69(57%) 1:51(43%)
Density Based Clustering	63.29%	64.77%	0.05 sec	0: 63(53%) 1: 57(48%)
Hierarchical Clustering	61.70%	63.29%	0.23 sec	0:120(100%) 1: 0 (0%)

Table 5. Comparison Between Various Algorithms for the number of clusters k=3.

Algorithms	Accuracy	F-Measure	Time Taken	Cluster Instances
K-Means	66.38%	73.2%	0.61 sec	0: 39(33%) 1: 15 (13%) 2: 66(55%)
K-Medoid	75.49%	79.3%	0.07 sec	0:66(57%) 1:39 (33%) 2:15(13%)
Density Based Clustering	61%	64.77%	0.32 sec	0: 50(42%) 1: 14(12%) 2: 56(47%)
Hierarchical Clustering	62.69%	64.73%	0.54 sec	0:120(100%) 1: 0 (0%) 2: 0 (0%)

Table 6. Comparison Between Various Algorithms for the number of clusters k=4.

Algorithms	Accuracy	F-Measure	Time Taken	Cluster Instances
K-Means	62.1%	75.2%	0.376 sec	0: 9(8%) 1: 15 (13%) 2: 66(55%) 3:30(25%)
K-Medoid	71.22%	83%	0.088 sec	0:64(53%) 1:39(33%) 2:6(5%) 3:11(9%)
Density Based Clustering	63.29%	65.97%	0.35 sec	0: 26(22%) 1: 14(12%) 2: 56(47%) 3:24(20%)
Hierarchical Clustering	61.70%	63.19%	0.97 sec	0:120(100%) 1: 0 (0%) 2: 0 (0%) 3: 0 (0%)

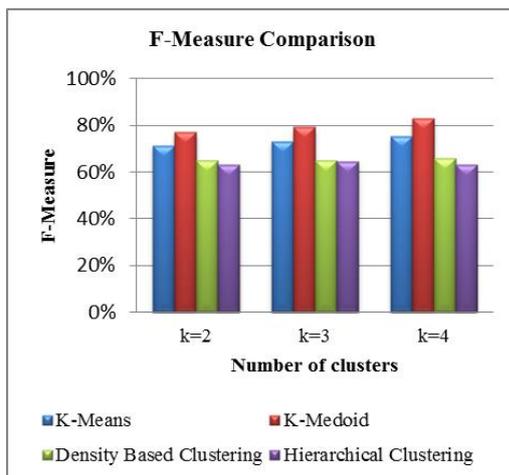


Fig.12. F-Measure comparison for Ionosphere dataset.

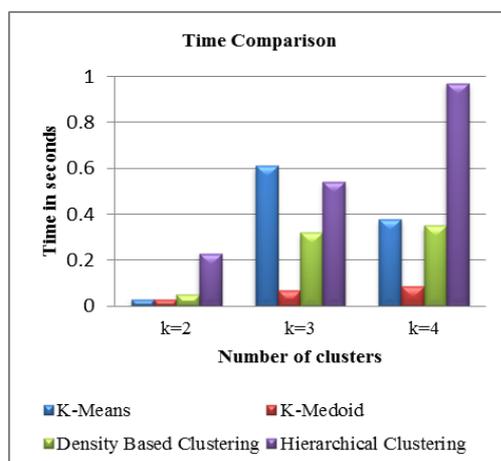


Fig.13. Time comparison for traffic dataset.

VII. CONCLUSION

In this paper, the focus is on data mining's algorithm ie. Clustering, a method of grouping data items in a cluster and to review its various algorithms like partitioning algorithms, hierarchical algorithms, and density based algorithms. These algorithms have their different pros

and cons. Partitioning algorithms, DBSCAN, and Hierarchical algorithms are analyzed using WEKA tool for Traffic email dataset among which K-Medoid tend to be a good algorithm. This algorithm has its own limitations but still, it fit to solve most of the clustering issues as there is no such algorithm that can solve all the clustering problems. K-Medoid algorithm is a simple algorithm which can generate arbitrary shapes of clusters rather than a particular shape, it can handle the large volume of data, K-Medoid can efficiently handle noise and outliers, and can work on both numeric and nominal data. The main purpose of this survey paper is to enlighten various clustering algorithms used for grouping the data items in the clusters. The survey demonstrated the good algorithm for clustering based on the factor like low execution time, high accuracy and F-Measure for personally collected Traffic dataset and ionosphere dataset available at UCI Repository. The survey provides the basic knowledge of WEKA tool. The paper starts by basic definition of clustering and listed all its possible algorithms for clustering. The paper concludes that partitioning algorithms are efficient algorithms as compared to other algorithms in clustering technique.

ACKNOWLEDGMENT

The author would like to express its cordial thanks to CT group of Engg., Mgmt. & Tech. for their motivational participation in the research field. The author is also grateful to computer science research group for the support. The author is thankful to its mentor for encouragement and guidance throughout the research work.

REFERENCES

[1] P. Verma and D. Kumar, "Association Rule Mining Algorithm's Variant Analysis," *Int. J. of Comput. App. (IJCA)*, vol. 78, no. 14, pp. 26–34, Sept. 2013. "DOI: 10.5120/13593-1366"

[2] R. Xu. and D. Wunsch, "Survey of Clustering Algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005. "DOI: 10.1109/TNN.2005.845141"

- [3] M. S. Chen, J. Han, and P. S. Yu, "Data mining: An Overview from a Database Perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 6, pp. 866–883, Dec.1996. "DOI: 10.1.1.106.8976"
- [4] A. Silberschatz, M. Stonebraker and J.D. Ullman, "Database Research: Achievements and Opportunities into the 21st Century," *Report NSF workshop Future of Database Systems Research*, May 1995. "DOI:10.1.1.47.8847"
- [5] A. Jain and R. Dubes, "Clustering Methodologies in Exploratory Data Analysis," *ELSEVIER- Advances in Computer*, vol. 19, pp. 113-228, Feb 2008. "DOI: [https://doi.org/10.1016/S0065-2458\(08\)60034-0](https://doi.org/10.1016/S0065-2458(08)60034-0)"
- [6] A. Baraldi and E. Alpaydin, "Constructive Feedforward ART clustering networks—Part I," *IEEE Trans. Neural Network.*, vol. 13, no. 3, pp. 645–661, May 2002. "DOI:10.1109/TNN.2002.1000130"
- [7] A. Baraldi and E. Alpaydin, "Constructive Feedforward ART clustering networks—Part II," *IEEE Trans. Neural Network.*, vol. 13, no. 3, pp. 662–677, May 2002. "DOI:10.1109/TNN.2002.1000131"
- [8] V. Cherkassky and F. Mulier, *Learning From Data: Concepts, Theory and Methods*, 2ND ED., New York: Wiley, 1998.
- [9] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2ND ED., New York: Wiley, 2001.
- [10] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a Review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sept. 1999. "DOI:10.1145/331499.331504"
- [11] M. S. B. PhridviRaj and C. V. GuruRao, "Data Mining – Past, Present and Future – A Typical Survey on Data Streams," in *7th Int. Conf. Interdis. in Eng. (INTER-ENG 2013)- ELSEVIER.*, vol. 12, pp. 255–263, Dec 2014. "DOI: 10.1016/j.procy.2013.12.483"
- [12] U. Fayyad, G. P. Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Mag.*, vol. 17, no. 3, pp. 1- 34, 1996.
- [13] P. Berkhin, "A Survey of Clustering Data Mining," *Springer - Group. Multidimens. Data*, pp. 25–71, 2006. "DOI: 10.1007/3-540-28349-8_2"
- [14] B. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, 5TH ED., London:Arnold, 2001.
- [15] A. Jain, A. Rajavat, and R. Bhartiya, "Design, analysis and implementation of modified K-mean algorithm for large data-set to increase scalability and efficiency," in *4th Int. Conf. Comput. Intell. Commun. Networks (CICN)* pp. 627–631, Dec. 2012. "DOI :10.1109/CICN.2012.95"
- [16] P. Chauhan and M. Shukla, "A Review on Outlier Detection Techniques on Data Stream by Using Different Approaches of K-Means Algorithm," in *International Conference on Advances in Computer Engineering and Applications (ICACEA)*, pp. 580–585, July 2015. "DOI :10.1109/ICACEA.2015.7164758".
- [17] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sept 1999." DOI:10.1145/331499.331504"
- [18] S. Firdaus and A. Uddin, "A Survey on Clustering Algorithms and Complexity Analysis," *Int. J. Comput. Sci. Issues (IJCSI)*, vol. 12, no. 2, pp. 62–85, March 2015.
- [19] D. Sisodia, "Clustering Techniques : A Brief Survey of Different Clustering Algorithms," *Int. J. latest trends Eng. Technology (IJLTET)*, vol. 1, no. 3, pp. 82–87, Sept. 2012.
- [20] K. N. Ahmed and T. A. Razak, "An Overview of Various Improvements of DBSCAN Algorithm in Clustering Spatial Databases," *Int. J. Adv. Res. Comput. Commun. Eng. (IJARCC)*, vol. 5, no. 2, pp. 360–363, 2016. "DOI: 10.17148/IJARCC.2016.5277"
- [21] A. Joshi, R. Kaur "A Review : Comparative Study of Various Clustering Techniques in Data Mining," *Int. J. Adv. Res. Comput. Sci. Soft. Eng. (IJARCSSE)*, vol. 3, no. 3, pp. 55–57, March 2013.
- [22] A. Naik, "Density Based Clustering Algorithm," 06-Dec-2010.[Online].Available:<https://sites.google.com/site/dataclusteringalgorithms/density-based-clustering-algorithm>. [Accessed: 15-Jan-2017].
- [23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10-18, June 2009. "DOI: 10.1145/1656274.1656278"
- [24] R. Ng and J. Han , " Efficient and Effective Clustering Method for Spatial Data Mining," in *20th VLDB Int. Conf. on Very Large Data Bases* , pp. 144-155, Sept. 1994.
- [25] Cios, K. J., W. Pedrycz, et al., *Data Mining Methods for Knowledge Discovery*, vol. 458, Springer Science & Business Media, 2012. " DOI: 10.1007/978-1-4615-5589-6"
- [26] S. Dixit, and N. Gwal, "An Implementation of Data Pre-Processing for Small Dataset," *Int. J. of Comp. App. (IJCA)*, vol. 10, no. 6, pp. 28-3, Oct. 2014. "DOI: 10.5120/18080-8707"
- [27] S. Singhal and M. Jena, "A Study on WEKA Tool for Data Preprocessing , Classification and Clustering," *Int. J. Innov. Technol. Explor. Eng.*, vol. 2, no. 6, pp. 250–253, May 2013. "DOI:10.1.1.687.799"
- [28] O. Y. Alshamesti, and I. M. Romi, "Optimal Clustering Algorithms for Data Mining" *Int. Journal of Info. Eng. and Electron. Bus. (IJIEEB)*, vol.5, no.2 ,pp. 22-27, Aug 2013. "DOI: 10.5815/ijieeb.2013.02.04 "
- [29] N. Lekhi, M. Mahajan "Outlier Reduction using Hybrid Approach in Data Mining," *Int. J. Mod. Educ. Comput. Sci.*, vol. 7, no. 5, pp. 43–49, May 2015. "DOI: 10.5815/ijmecs.2015.05.06"
- [30] C. L. P. Chen and C.Y. Zhang, "Data- Intensive Applications, Challenges, Techniques and Technologies: A survey on Big Data." *ELSEVIER- Inform. Sci.*, pp. 314-347, Aug. 2014. "DOI: 10.1016/j.ins.2014.01.015"
- [31] E. Rahm, and H. H. Do , "Data cleaning: Problems and current approaches." *IEEE- Data Eng. Bull.*, vol. 23, no. 4, pp. 3-13, Dec 2000. "DOI:10.1.1.101.1530"

Authors' Profiles



Harjot Kaur was born in Jalandhar, Punjab, India in 1992. She received the B.Tech degree in Computer Science and Engineering from C.T. Group of Institution, Jalandhar, India, in 2014. She is currently a student of M.Tech in Computer Science and Engineering from C.T. Group of Institution, Jalandhar, India. The M.Tech degree will be completed in 2017. Her main areas of research interests are Data Mining, Data Warehousing.



Prince Verma, he received the B.Tech degree in Computer Science from MIMIT, Malout (Pb), India in 2008 and M.Tech degree in Computer Science in 2013 from DAVIET, Jalandhar (Pb), India. Currently, he is Assistant Professor in Computer Science Department of CTIEMT, Jalandhar (Pb), India. His research focuses on Data

Mining, Algorithm optimization techniques.

How to cite this paper: Harjot Kaur, Prince Verma,"Comparative Weka Analysis of Clustering Algorithm's", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.8, pp.56-67, 2017. DOI: 10.5815/ijitcs.2017.08.07