Research Journal of Applied Sciences, Engineering and Technology 11(1): 27-39, 2015
DOI: 10.19026/rjaset.11.1672
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: December 14, 2014 Accepted: April 13, 2015 Published: September 05, 2015

Research Article

Privacy Preserving Probabilistic Possibilistic Fuzzy C Means Clustering

[1]V.S. Thiyagarajan and [2]Venkatachalapathy
[1]Annamalai University, Chidhambaram, India
[2]Department of Computer Science and Engineering, Faculty of Engineering and Technology,
Annamalai University, Chidhambaram, India

Abstract: Due to this uncontrollable growth of data, clustering played major role to partition into a small sets to do relevant processes within the small sets. Recently, the privacy and security are extra vital essentials when data is large and the data is distributed to other sources for various purposes. According to that, the privacy preservation should be done before distributing the data. In this study, our proposed algorithm meets the both requirements of achieving the clustering accuracy and privacy preserving of the data. Initially, the whole dataset is divided to small segments. The next step is to find the best sets of attributes combinations, which are attained through, attribute weighing process, which leads to attain the privacy preservation through vertical partitioning. The next is to apply the proposed Probabilistic Possibilistic Clustering Algorithm (PPFCM) for each segment, which produces the number of clusters for each segment. The next step is applying the PPFCM on the centroids of the clusters. The corresponding data tuples of the grouped centroids join to attain the final clustered result. The implementation is done using JAVA and the performance of the proposed PPFCM algorithm is compared with possibilistic FCM and probability-clustering algorithm for the benchmark datasets.

Keywords: Clustering, possibilistic fuzzy C means clustering, privacy preserving, probabilistic clustering

INTRODUCTION

Clustering (Ji *et al*., 2012; Das *et al*., 2008; Chen *et al*., 2011; Januzaj *et al*., 2004; Jin *et al*., 2006; Li *et al*., 2012; Patel *et al*., 2013; Roy, 2014) has been considered widely for more than forty years in data mining arena (Osmar, 1999; Mehmed, 2003; Kusiak and Smith, 2007) and over several application in many fields because of its wide applications. Clustering is the strategy of engaging data entities into a set of separate groups called clusters so that protests in each one group are more identical to one another than objects from various clusters. The literary works represents with a huge number of strategies for effective grouping of information. These strategies could be classified into nearest neighbour clustering, fuzzy clustering, partition clustering, hierarchical clustering, artificial neural networks for clustering, statistical clustering algorithms, density-based clustering algorithm etc. In these techniques, hierarchical and partition clustering algorithms are two essential methodologies of expanding fervor toward exploration groups. Various hierarchical clustering strategies can normally realize reasonable clustering outcomes. In spite of the fact that the hierarchical clustering procedure is regularly depicted as a superior quality clustering approach, this method does not contain any procurement for the rearrangement of data, which may have been crudely grouped at the initial stage. Moreover, the majority of the hierarchical clustering strategies are computationally accelerated and entail high memory storage (Izakian *et al*., 2009).

Currently, huge data clustering has been broadly analysed over in several fields, including statistics, machine learning, pattern recognition and image processing (Ester *et al*., 1996; Ng and Han, 1994; Zhang *et al*., 1996b). In these applications, the versatility of clustering routines and the strategies for huge data clustering much dynamic exploration has been committed. To conquer the issues happened in vast database grouping distinctive routines has been presented, including instatement by clustering a model of the information and employing a primary coarse partitioning of the whole data set (Wehrens and Buydens, 2004). On the other hand, the most conspicuous delegates are partitioning clustering systems, for example, CLARANS (Ng and Han, 1994); hierarchical clustering strategies, for example, BIRCH (Zhang *et al*., 1996a) grid clustering algorithms, for example, Sting (Wang *et al*., 1997) and Wavecluster (Sheikholeslami *et al*., 1998). Every technique has its focal points and weaknesses. They are not suitable for handling substantial databases. It is hard to secure both high precision and productivity in a clustering strategy of expansive data. The two targets dependably clash with one another. To process huge information sets, the

Corresponding Author: V.S. Thiyagarajan, Annamalai University, Chidhambaram, India control of a solitary computer is insufficient. Parallel and distributed clustering is the significant scheme. It is very versatile and ease to perform clustering in a distributed surroundings.

Because of the colossal application of clustering strategy to the substantial data, it has eminent issues about securing data against unapproved access is an imperative objective of database security and protection. Privacy is a term which is connected with this a mining assignment so we can conceal some pivotal data which we would prefer not to expose to general society (Jain *et al*., 2011). Organizations and different associations frequently need to distribute micro data, e.g., medical information or enumeration information, for analysis and different resolutions. Commonly, such information is put away in a table and each one record (row) relates to one specific individual. Each one record has various characteristics, which could be partitioned into the accompanying three classifications:

- Attributes that unmistakably distinguish individuals
- Attributes whose qualities might be known from different sources
- Attributes that are viewed as delicate

However discharging micro data gives valuable data to specialists, it displays revelation risk to the people whose information are present in the table (Islam and Brankovic, 2004). Individual records are frequently thought to be private and delicate. Three sorts of data revelation have been recognized in the novel works, identity disclosure, attribute disclosure and membership disclosure.

The main objective of the study is to design and develop an algorithm for privacy preserving probabilistic possibilistic clustering algorithm. The main contribution of the research is to attain the privacy preserving and better clustering accuracy. Initially, the whole dataset is divided to small segments. The next step is to find the best sets of attributes combinations, which are attained through, attribute weighing process, which leads to attain the privacy preservation through vertical grouping of attributes. The next is to apply the proposed Probabilistic Possibilistic Clustering algorithm (PPFCM) for each segment, which produces the number of clusters for each segment. The next step is apply the PPFCM on the centroids of the clusters. The corresponding data tuples of the grouped centroids joined together to attain the final clustered result.

MATERIALS AND METHODS

In this study, we achieving the privacy preserving through combining the relative attributes together. The relations between the attributes are obtaining through attribute weighting algorithm, which computes the relation score value of the attributes. Based on the computed score value the attributes are combined together with the intention of attain the privacy preserving of the data. In order to improve the clustering accuracy, in this study, we combine the probabilistic clustering algorithm (Lyigun, 2008) with possibilistic fuzzy c means clustering algorithm (Pal *et al*., 2005) and we derived the Probabilistic Possibilistic Fuzzy C Means clustering algorithm (PPFCM). With the intention of reduce the running time of the clustering process in this study, we divide the whole database into S number of segments, subsequently we apply our proposed clustering algorithm on each segmentation. Once the segments are clustered through our proposed PPFCM clustering algorithm subsequently, we collect the centroids of the clustered segments, which are subjected to PPFCM clustering process. The corresponding data of each centroid of the group is combined together to get final clustered data. The following figure represents the overall architecture of the proposed privacy preserving clustering algorithm.

Figure 1 represents the entire architecture of the proposed method. At first, the whole database is divided into S number of segments. For each segment, we evaluate the combined attributes through the calculation of weighted holoentropy of each attribute combination. Once the calculation of weighted holoentropy of the attribute combinations are finished, we select the best set of attributes and the data of the selected attributes are combined together for each segment. Now the attributes are Reduced M into (M-R) where value of R represents the number of reduced attributes. Once the attribute combination process is over, the next step is to apply our proposed Probabilistic Possibilistic Fuzzy-C-Means clustering algorithm (PPFCM) for each segment. The result of our proposed PPFCM returns the clustered data for each segment. Next, the algorithm selects the centroids of the clustered data of the every segmentation in order to apply on PPFCM. Once the selected centroids are clustered, the corresponding data also grouped together to get the final clustering.

Segmentation: We designed our proposed methodology to achieve the better clustering accuracy with less computation time. In order to achieve that in this study, the whole database divides into *S* number of segments. The segmentation process is leads to reduce the running time of the clustering process. The construction of segmentation is based on the number of tuples and there is no change in terms of attributes of the database during the segmentation process.

The database $DB = N \times M$, where N represents total number of tuples in the database and M represents the total number of attributes. Each of the segmentation is made based on splitting the number of tuples. Each of

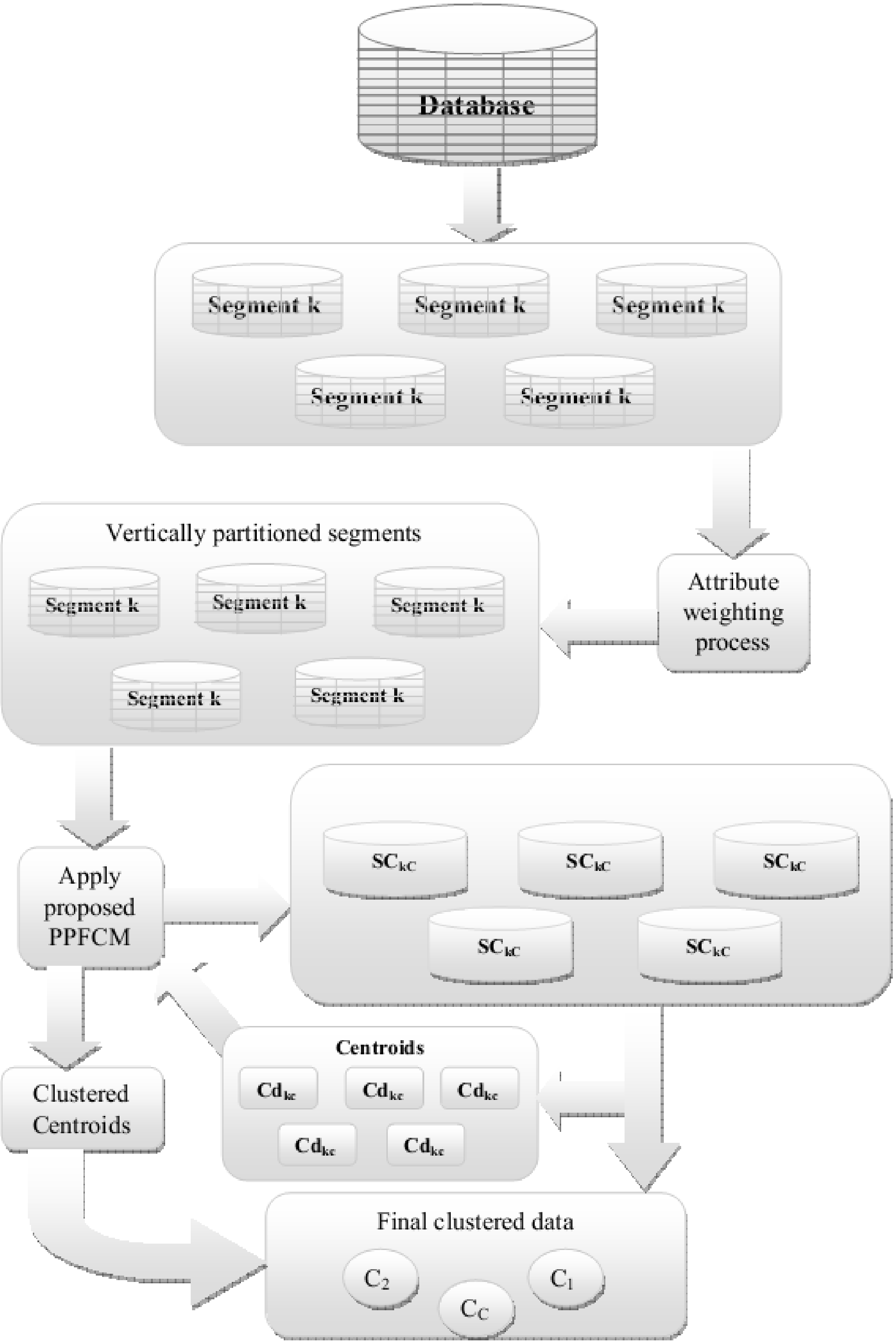

Fig. 1: Architecture of proposed privacy preserving clustering methodology

the segmentation is represented by $s_k = [n_i \times M]$, where the values k varies from one to S. Consider the database $DB = 100\times10$, where the value of 100 represents the number of tuples of the database and the value of 10 represents the number of attributes present in the database. Let us consider the database splits into four segments then, the size of the divided segments are $s_1 = 25\times10$, $s_2 = 25\times10$, $s_3 = 25\times10$, $s_4 = 25\times10$.

Attaining privacy through attribute combination: In this study, we attain the privacy through attribute combination. Combining of attributes is not an easy task since the database consist of many attribute they are irrelevant with one another. For that reason, we combine every attributes and calculate the weighted holoentropy for each combination. From the resultant value of weighted holoentropy of the combination is evaluated and the best set of attributes are combined together to preserve the privacy of the data. The attribute combination process is initiated with the process of generating the attribute combination with the length of two. From each s_k, we generate the set of attribute combination with the length of two. The attribute combination of the segmentation can be indicated as $C_k = \{c^k_v\}$ where the value of *v* varies from one to *V* and the value of *V* can be obtaining through the following Eq. (1) where the value of M represents the maximum number of attribute of the database DB:

$$V = \sum_{j=1}^{M-1} \left(M - j\right) \tag{1}$$

Once the attribute combinations are generated for each segment, the next step is to evaluate those combinations through weighted holoentropy. In order calculate the weighted holoentropy initially, the next step is to find the set of unique items from each combination $c^k_v = \{UI_t^{vk}\}$ where, value of t varies from one to *T* and the value of *v* indicates the combination id and the value of *k* signifies segmentation id. The calculation of weighted holoentropy (Wu and Wang, 2013) of the attribute combination c^k_v is attaining through the following Eq. (2) to (4):

$$H\left(UI_t^{vk}\right) = -p\left(UI_t^{vk}\right)\log\left(p\left(UI_t^{vk}\right)\right) \tag{2}$$

$$w\left(UI_t^{vk}\right) = 2\left(1 - \left(\frac{1}{1 + \exp\left(H\left(UI_t^{vk}\right)\right)}\right)\right) \tag{3}$$

$$w\left(c_v^k\right) = \sum_{t=1}^{T} w\left(UI_t^{wk}\right) \times H\left(UI_t^{vk}\right) \tag{4}$$

Once the calculation of weighted holoentropy of attribute combination for each segment is finished, the next step finds the overall weighted holoentropy of the attribute combination, which is calculated through the following Eq. (5):

$$W\left(c_v\right) = \sum_{k=1}^{S} w\left(c_v^k\right) \tag{5}$$

Once we calculated the overall weighted holoentropy of the every attribute combination, the next step is to arrange attribute combinations in descending order based on the values of weighted holoentropy. The next step is the selection of attribute combination based on maximum value of weighted holoentropy with one condition, which says that the selected attributes in combinations should not repeat with the other combination. For example consider the set of sorted attribute combinations (a_1, a_3), (a_1, a_2), (a_1, a_4), (a_1, a_5), (a_2, a_3). (a_2, a_4), (a_2, a_5), (a_3, a_4), (a_3, a_5), (a_4, a_5), where (a_1, a_3) is the attribute combination selected first for attribute combining process which has more weighted holoentropy value, then the next removing the attribute combinations which contains the attribute a_1 or a_3 from the sorted list. The modified sorted list contains the combinations are (a_2, a_4), (a_2, a_5), (a_4, a_5) from which the one combination set is selected which maximum value. This process is repeated until the modified sorted list has no combinations.

As per the selected set of combinations, the corresponding attributes are combined together for each segment and called as sanitized segment, which is denoted by $ss_k = [n_i \times m_j]$ subsequently every sanitized segments are subjected to our proposed Probabilistic Possibilistic Fuzzy C Means clustering algorithm (PPFCM).

Probabilistic Possibilistic Fuzzy C Means clustering (PPFCM): Initially, the number of Cluster (C) is defined by the user, which is similar for every segment. Once the number of cluster is decided, the next step is to computation of distance between the centroids and data tuples for each segment.

Calculation of distance matrix: In this study, we utilized Euclidian distance function Eq. (6) to calculate the distance between centroids and data tuples to compute the distance matrix. For each segmentation, we calculate the distance matrix:

$$d\left(d_i, v_j\right) = \sqrt{\sum_{i,j=1}^{i=n,\, j=n} \left(d_i - v_j\right)^2} \tag{6}$$

Figure 2 represents the distance matrix of sanitized segment *D* (ss_k) where the value of d_{ij} represents the distance of i^th^ data tuple with respect to the j^th^ centroid. Likewise, the distance function is used to calculate the distance between the every data tuple with every centroid value and finally the distance matrix is generated which is represented in the Fig. 2.

Calculation of probability matrix: Once the computation of distance matrix for each segmentation is over, the next step is to build the probability matrix. The probability matrix has values of probability of data

$$D(ss_k)=\begin{bmatrix} d_{11} & d_{12} & d_{1j} & d_{1C-1} & d_{1C} \\ d_{21} & d_{22} & d_{2j} & d_{2\,C-1} & d_{2C} \\ d_{i1} & d_{i2} & d_{i\,j} & d_{i\,C-1} & d_{i\,C} \\ d_{n-11} & d_{n-12} & d_{n-1\,j} & d_{n-1\,C-1} & d_{n-1\,C} \\ d_{n1} & d_{n2} & d_{n\,j} & d_{n\,C-1} & d_{n\,C} \end{bmatrix}$$

Fig. 2: Represents the distance matrix

$$P(ss_k)=\begin{bmatrix} p_{11} & p_{12} & p_{1j} & p_{1C-1} & p_{1C} \\ p_{21} & p_{22} & p_{2j} & p_{2\,C-1} & p_{2C} \\ p_{i1} & p_{i2} & p_{i\,j} & p_{i\,C-1} & p_{i\,C} \\ p_{n-11} & p_{n-12} & p_{n-1\,j} & p_{n-1\,C-1} & p_{n-1\,C} \\ p_{n1} & p_{n2} & p_{n\,j} & p_{n\,C-1} & p_{n\,C} \end{bmatrix}$$

Fig. 3: Represents the probability matrix

$$T(ss_k)=\begin{bmatrix} t_{11} & t_{12} & t_{1j} & t_{1C-1} & t_{1C} \\ t_{t21} & t_{22} & t_{2j} & t_{2\,C-1} & t_{2C} \\ t_{i1} & t_{i2} & t_{i\,j} & t_{i\,C-1} & t_{i\,C} \\ t_{n-11} & t_{n-12} & t_{n-1\,j} & t_{n-1\,C-1} & t_{n-1\,C} \\ t_{n1} & t_{n2} & t_{n\,j} & t_{n\,C-1} & t_{n\,C} \end{bmatrix}$$

Fig. 4: Represents the typicality matrix

$$U(ss_k)=\begin{bmatrix} u_{11} & u_{12} & u_{1j} & u_{1C-1} & u_{1C} \\ u_{t21} & u_{22} & u_{2j} & u_{2\,C-1} & u_{2C} \\ u_{i1} & u_{i2} & u_{i\,j} & u_{i\,C-1} & u_{i\,C} \\ u_{n-11} & u_{n-12} & u_{n-1\,j} & u_{n-1\,C-1} & u_{n-1\,C} \\ u_{n1} & u_{n2} & u_{n\,j} & u_{n\,C-1} & u_{n\,C} \end{bmatrix}$$

Fig. 5: Represents the membership matrix

tuple with respect to each centroid, which is derived from Lyigun (2008). The probability value is computed through the following Eq. (7):

$$p_{i\,j}=\frac{\sum_{k=1}^{C}e^{d_{i\,k}}}{\sum_{l=1}^{C}e^{d\,l}} \qquad (k\neq j) \tag{7}$$

With the help of the above Eq. (6), the probability value of each data tuple with respect to every centroid is completed then the probability matrix get generated which is displayed in the following Fig. 3.

Figure 3 represents the probability matrix of sanitized segment *P* (*ss$_k$*) where the value of p_{ij} represents the probability of ith data tuple chance to go for the jth centroid. Likewise, the distance function is used to calculate the distance between the every data tuple with every centroid value and finally the distance matrix is generated which is represented in the Fig. 2.

Calculation of typicality matrix: Once we calculated the probability matrix, the next step is to compute the typicality matrix. This typicality matrix is derived from Pal *et al.* (2005). With the help of the following Eq. (8), the probability value of each data tuple with respect to every centroid is completed then the probability matrix is generated which is displayed in the following Fig. 4:

$$t_{i\,j}=\frac{1}{\sum_{j=1}^{n}\left(\frac{d_{i\,k}}{d_{i\,j}}\right)} \qquad 1\leq i\geq n, 1\leq k\geq C \tag{8}$$

Figure 4 represents the probability matrix of sanitized segment *T* (*ss$_k$*) where the value of t_{ij} represents the probability of ith data tuple chance to go for the jth centroid. Likewise, the distance function is used to calculate the distance between the every data tuple with every centroid value and finally the distance matrix is generated which is represented in the Fig. 2.

Calculation of membership matrix: The computation of membership matrix *U* (*ss$_k$*) is made with the help of computation of membership value of data tuple which is calculated using the following Eq. (9) where the value of d_{ij} represents the distance of ith data tuple with respect to the jth centroid. Figure 5 represents the membership matrix. The value of e^{dij} represents the exponential value of d_{ij} and the p_{ij} indicates the probability of d_{ij}. The clustering the data tuple is made with respect to the membership value of the data tuple:

$$u_{i\,j}=\frac{1}{\sum_{k=1}^{C}\left(\frac{(p_{i\,j})^{2}\left(e^{d_{i\,j}}\right)+\left(d_{i\,j}\right)}{d_{i\,k}}\right)} \qquad 1\leq i\geq n, 1\leq j\geq C \tag{9}$$

Updation of centroid: Once the clusters are made, the next step is to update the centroids based on the following Eq. (10):

$$v_{j}^{l}=\frac{\sum_{i=1}^{n}\left(u_{i\,j}+t_{i\,j}\right)x_{i}}{\sum_{i=1}^{n}\left(u_{i\,j}+t_{i\,j}\right)} \qquad 1\leq j\geq C \tag{10}$$

Once the centroids are updated for the every segment, the next step is to begin with the process of computing the distance with the newly updated centroids and it continues up to computation of updation of centroids. This process is repeated until the updated centroids of each segment are similar in consecutive iterations.

Algorithm procedure:
Input: Database
Output: Clustered data
Parameters:

DB = Database $N×M$
S = Total number of segments
s_k = Segmentation $s_k = [n_i×M]$
C_k = Set of combination of s_k, $C_k = \{c^k{}_v\}$
$c^k{}_v$ = Combination of s_k
V = Total number of combination in s_k
$\{UI_t^{vk}\}$ = Set of unique items of $c^k{}_v$
$H\,(UI_t^{vk})$ = Holoentropy of UI_t^{vk}
$w\,(UI_t^{vk})$ = Weighted holoentropy of UI_t^{vk}
$w\,(c_v^k)$ = Weighted holoentropy $c^k{}_v$
$W\,(c_v)$ = Weighted holoentropy of combination c_v
ss_k = Sanitized segment
C = Total number of clusters
$\{v_j\}$ = Set of centroids
d_{ij} = Distance of ith data tuple with respect to the jth centroid
$D\,(ss_k)$ = Distance matrix of ss_k
p_{ij} = Probability of d_{ij}
$P\,(ss_k)$ = Probability matrix of ss_k
t_{ij} = Typicality value of d_{ij}
$T\,(ss_k)$ = Typicality matrix of ss_k

Pseudo code:

```
Begin
    Read $DB$
      Get $S$
        Split $DB$ into $s_k$
          For each $s_k$
            Call sanitization
              Call PPFCM
          End for
              Select centroids
                Call PPFCM
                  For each group
                    Concatenate data points
                  End for
End
```

Subroutine: Sanitization

```
Compute $C_k$
    For each $c^k{}_v$
        Generate $\{UI_t^{vk}\}$
          For each $UI_t^{vk}$
            Compute $H\,(UI_t^{vk})$ Eq. (2)
              Compute $w\,(UI_t^{vk})$ Eq. (3)
    End for
              Compute $w\,(c_v^k)$ Eq. (4)
  End for
Compute $W\,(c_v)$ Eq. (5)
  Find best $\{c_v\}$
    For each $s_k$
      $ss_k$ ← Concatenate best $\{c_v\}$
End for
```

Subroutine: PPFCM

```
Get $C$
    Select $\{V_v\}$
      Compute $d_{ij}$ Eq. (6)
Construct $D\,(ss_k)$
    Compute $p_{ij}$ Eq. (7)
        Construct $P\,(ss_k)$
          Compute $t_{ij}$ Eq. (8)
            Construct $T\,(ss_k)$
              Compute $t_{ij}$ Eq. (9)
                Construct $U\,(ss_k)$
        Construct clusters based on $U\,(ss_k)$
              Update $v^l_j$ Eq. (10)
                If all $v_j^l == v_j^{l+1}$
              Terminate
        Else
Go to compute $d_{ij}$.
```

RESULTS AND DISCUSSION

This section presents the results obtained from the experimentation and its detailed discussion about the results. The proposed approach of PPFCM is experimented with the Adult Datasets and mushroom dataset. The result is evaluated with the probabilistic clustering (Lyigun, 2008) and possibilistic fuzzy c means clustering (Pal *et al.*, 2005), accuracy and computation time.

Dataset description: In this study, we utilized Adult dataset UCI 1994 (Adult dataset, 1994) and Mushroom dataset UCI 1981 (Mushroom dataset, 1981). are obtained from UCI machine learning repository. The descriptions of the above datasets are given in the following Table 1.

Evaluation measure: Clustering accuracy is used to evaluate the clustering algorithm through counting the number of data exactly assigned data, which is calculated through the following Eq. (11):

$$Clustering\ accuracy = \frac{1}{N}\sum_k \max_j \left| w_k \cap c_j \right| \tag{11}$$

Table 1: Dataset description

Name of the dataset	Number of instances	Number of attributes
Adult UCI 1994	48842	14
Mushroom UCI 1981	8124	22

The above Eq. (11) represents the calculation of accuracy of the resultant clustered data, which is done through, compared the original class data where W = (w_1, w_2,…, w_k) is resultant clustered data of the clustering algorithm and C = (c_1, c_2,…, c_J) is set of classes of the dataset used for clustering.

Performance evaluation based on number of clusters: In this section, we evaluate our proposed clustering algorithm based on running time and clustering accuracy by varying number of clusters. The above evaluation methodology will applied on both adult and mushroom dataset. Here, the numbers of clusters are varied from two to five and the number of data used for clustering is fixed as 2500.

Evaluation of running time: Figure 6 represents the running time of possibilistic clustering algorithm, probabilistic clustering algorithm and proposed PPFCM clustering algorithm. By analyzing the Fig. 6, when the number of clusters increased, the required running time clustering algorithms used for evaluation. Additionally, running time of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters however, the running time of the probabilistic clustering algorithm is higher than the proposed PPFCM for every number of clusters. From the Fig. 6, the minimum execution cost is attained by proposed PPFCM is 32457 milliseconds for number of clusters as two and the maximum execution cost is attained by proposed PPFCM is 47457 msec for number of clusters as five.

From the Fig. 7, when the number of clusters increased, the required running time of the clustering process also increased for three clustering algorithms used for evaluation. Additionally, running time of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters however, the running time of the probabilistic clustering algorithm is higher than the proposed PPFCM for every number of clusters. From the Fig. 7, the minimum execution cost is attained by proposed PPFCM is 58774 msec for number of clusters as two and the maximum execution cost is attained by proposed PPFCM is 82145 msec for number of clusters as five.

Evaluation of accuracy: Figure 8 represents the accuracy of possibilistic clustering algorithm, probabilistic clustering algorithm and proposed PPFCM clustering algorithm. By analysing the Fig. 8, when the number of clusters increased, the accuracy of clustering process is decreased gradually for three clustering algorithms as used for evaluation process. In Addition, accuracy of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters, which means that possibilistic-clustering algorithm performed well than the probabilistic clustering algorithm in terms of clustering accuracy however, accuracy of the proposed PPFCM algorithm is outperformed than the possibilistic clustering algorithm for every number of clusters. From the Fig. 8, the maximum accuracy is attained by proposed PPFCM is 81.9% for number of clusters as two and the minimum accuracy is attained by proposed PPFCM is 76% for number of clusters as five. The average accuracy of the proposed PPFCM, possibilistic FCM and probabilistic clustering algorithm are 80.55, of the clustering process also increased for three 77.98 and 77.1 respectively. The performance clearly

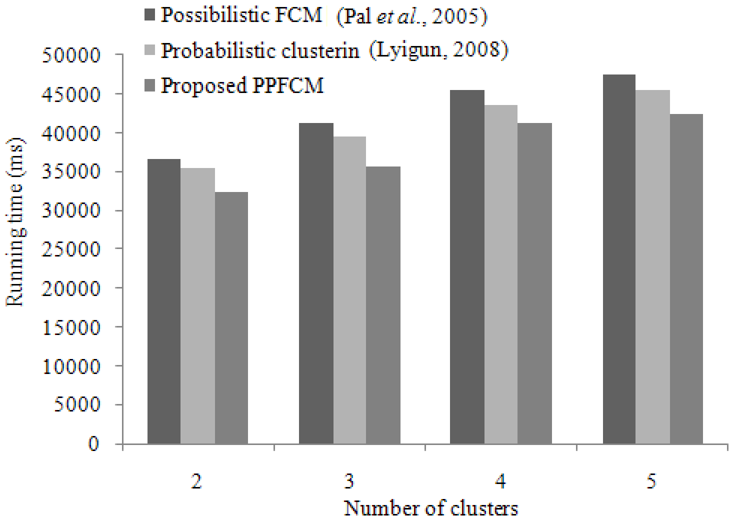

Fig. 6: Comparative analysis of running time of adult dataset based on number of clusters

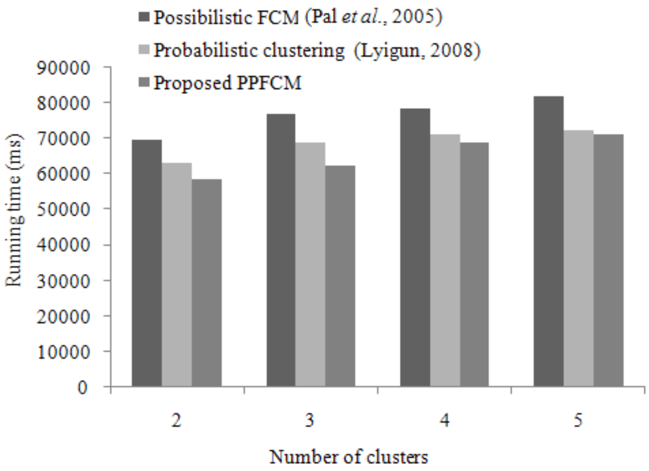

Fig. 7: Comparative analysis of running time of mushroom dataset based on number of clusters

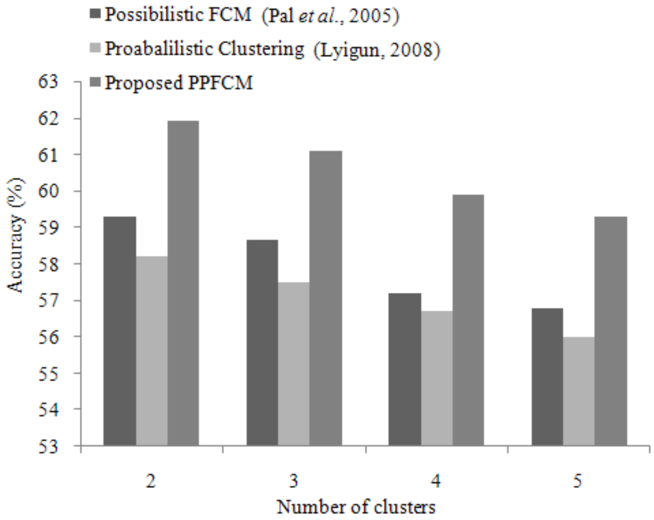

Fig. 8: Comparative analysis of clustering accuracy of adult dataset based on number of clusters

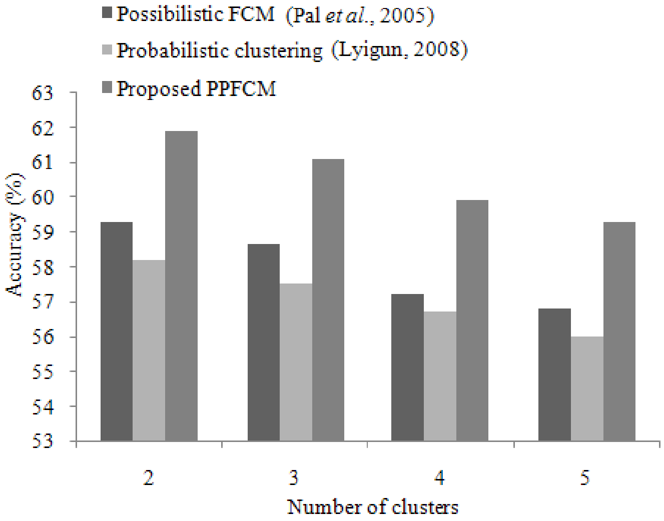

Fig. 9: Comparative analysis of clustering accuracy of mushroom dataset based on number of clusters

shows that the proposed PPFCM clustering algorithm outperformed than the existing probabilistic clustering algorithm and possibilistic FCM clustering algorithm in terms of accuracy.

Figure 9 represents the accuracy of possibilistic clustering algorithm, probabilistic clustering algorithm and proposed PPFCM clustering algorithm. By analyzing the Fig. 9, when the number of clusters increased, the accuracy of clustering process is decreased gradually for three clustering algorithms as used for evaluation process. In Addition, accuracy of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters, which means that possibilistic-clustering algorithm performed well than the probabilistic clustering algorithm in terms of clustering accuracy however, accuracy of the proposed PPFCM algorithm is outperformed than the possibilistic clustering algorithm for every number of clusters. From the Fig. 9, the maximum accuracy is attained by proposed PPFCM is 61.9% for number of clusters as two and the minimum accuracy is attained by proposed PPFCM is 56% for number of clusters as five. The average accuracy of the proposed PPFCM, possibilistic FCM and probabilistic clustering algorithm are 60.55, 57.98 and 57.1, respectively. The performance clearly shows that the proposed PPFCM clustering algorithm outperformed than the existing probabilistic clustering algorithm and possibilistic FCM clustering algorithm in terms of accuracy.

Performance evaluation based on number of data: In this section, we evaluate our proposed clustering algorithm based on running time and clustering accuracy by varying number data given for clustering process. The above evaluation methodology will applied on both adult and mushroom dataset. Here, the number of data given for clustering is 500, 1000, 1500, 2000 and 2500, respectively and the number of clusters is fixed value as five.

Evaluation of running time: Figure 10 represents the running time of possibilistic clustering algorithm, probabilistic clustering algorithm and proposed PPFCM clustering algorithm for the adult dataset. By analyzing the Fig. 10, when the number of data given for clustering is increased, the required running time of the clustering process also increased gradually for every clustering algorithm as used for evaluation purpose. Additionally, running time of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters however, the running time of the proposed PPFCM algorithm performed better than the probabilistic clustering for every number of clusters in terms of execution cost. From the Fig. 10, the minimum execution cost is attained by proposed PPFCM is 14578 msec for number of data given for the clustering process is 500 and the maximum execution cost is attained by proposed PPFCM is 47457 msec for the number of data given for the clustering process is 2500.

Figure 11 represents the running time of possibilistic clustering algorithm, probabilistic clustering algorithm and proposed PPFCM clustering algorithm for the adult dataset. By analyzing the Fig. 11, when the number of data given for clustering is increased, the required running time of the clustering process also increased gradually for every clustering algorithm as used for evaluation purpose. Additionally, running time of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters however, the running time of the

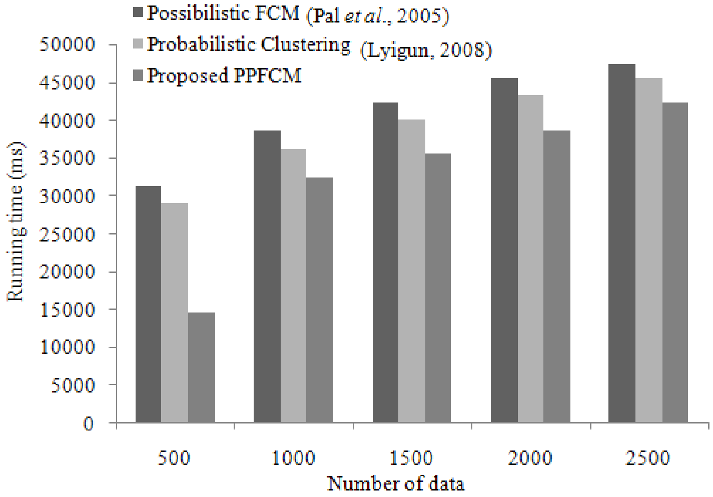

Fig. 10: Comparative analysis of running time of adult dataset based on number of clusters

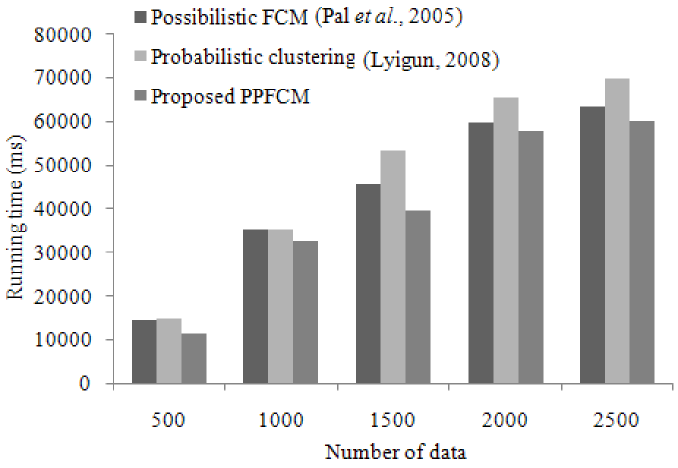

Fig. 11: Comparative analysis of running time of mushroom dataset based on number of clusters

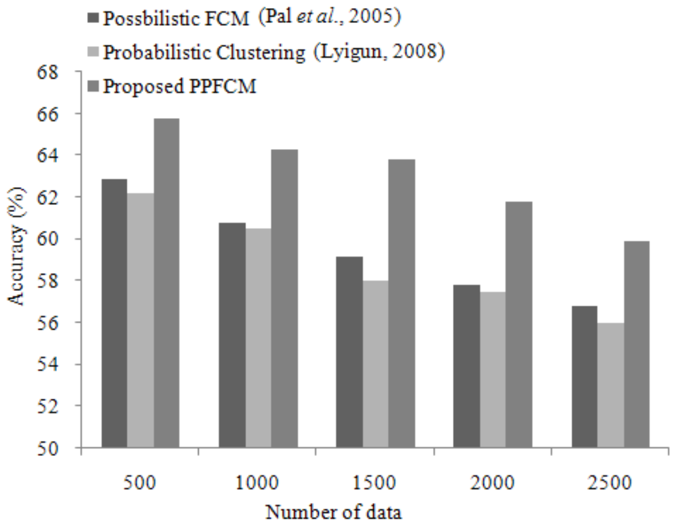

Fig. 12: Comparative analysis of clustering accuracy of adult dataset based on number of clusters

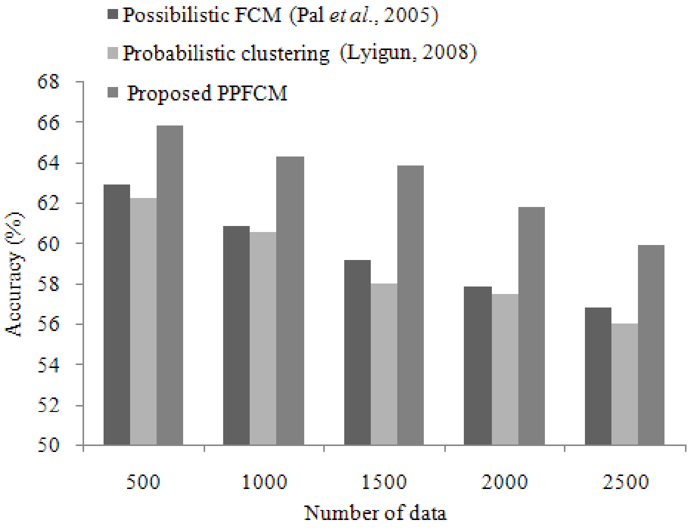

Fig. 13: Comparative analysis of clustering accuracy of mushroom dataset based on number of clusters

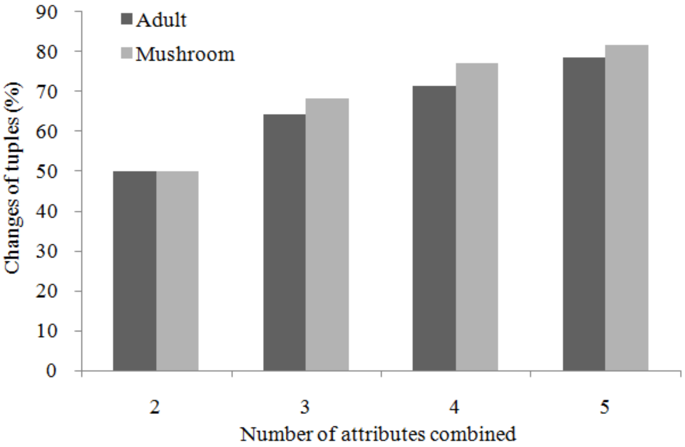

Fig. 14: Comparative analysis of privacy preserving of proposed PPFCM for mushroom dataset based on number of clusters

proposed PPFCM algorithm performed better than the probabilistic clustering for every number of clusters in terms of execution cost. From the Fig. 11, the minimum execution cost is attained by proposed PPFCM is 11224 msec for number of data given for the clustering process is 500 and the maximum execution cost is attained by proposed PPFCM is 69663 msec for the number of data given for the clustering process is 2500.

Evaluation of accuracy: Figure 12 represents the accuracy of possibilistic clustering algorithm, probabilistic clustering algorithm and proposed PPFCM clustering algorithm. By analyzing the Fig. 12, when the number of clusters increased, the accuracy of clustering process is decreased gradually for three clustering algorithms as used for evaluation process. In Addition, accuracy of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters, which means that possibilistic-clustering algorithm performed well than the probabilistic clustering algorithm in terms of clustering accuracy however, accuracy of the proposed PPFCM algorithm is outperformed than the possibilistic clustering algorithm for every number of clusters. From the Fig. 12, the maximum accuracy is attained by proposed PPFCM is 85.8% for number of data as 500 and the minimum accuracy is attained by probabilistic is 76% for number of data as 2500. The average accuracy of the proposed PPFCM, possibilistic FCM and probabilistic clustering algorithm are 83.116, 79.50 and 78.84, respectively. The performance clearly shows that the proposed PPFCM clustering algorithm outperformed than the existing probabilistic clustering algorithm and possibilistic FCM clustering algorithm in terms of accuracy.

Figure 13 represents the accuracy of possibilistic clustering algorithm, probabilistic clustering algorithm and proposed PPFCM clustering algorithm. By analyzing the Fig. 13, when the number of clusters increased, the accuracy of clustering process is decreased gradually for three clustering algorithms as used for evaluation process. In Addition, accuracy of the probabilistic clustering algorithm is lesser than possibilistic clustering algorithm for all number of clusters, which means that possibilistic-clustering algorithm performed well than the probabilistic clustering algorithm in terms of clustering accuracy however, accuracy of the proposed PPFCM algorithm is outperformed than the possibilistic clustering algorithm for every number of clusters. From the Fig. 13, the maximum accuracy is attained by proposed PPFCM is 65.8% for number of data as 500 and the minimum accuracy is attained by probabilistic clustering is 56% for number of data as 2500. The average accuracy of the proposed PPFCM, possibilistic FCM and probabilistic clustering algorithm are 63.116, 59.50 and 58.84, respectively. The performance clearly shows that the proposed PPFCM clustering algorithm outperformed than the existing probabilistic clustering algorithm and possibilistic FCM clustering algorithm in terms of accuracy.

Attain the privacy preserving: In this section, the following Fig. 14 represents the attaining of privacy preserving through our proposed methodology by combining the attributes efficiently, which leads to reduce the number of tuples. When the number of tuples reduced, the data become meaningless and the data privacy of the data maintained. The following Fig. 14 represents the changes on tuples in percentage made by proposed methodology. From Fig. 14, we attain the minimum 50% of privacy through combining the two attributes. The privacy of the data is increased when the number of attribute combination increase.

CONCLUSION

In this study, we developed an algorithm for privacy preserving probabilistic possibilistic clustering algorithm. The main contribution of the research is to attain the privacy preserving and better clustering accuracy. Initially, the whole dataset was divided to small segments. Subsequently the best sets of attributes combinations are attained through attribute weighing process, which leads to attain the privacy preservation through vertical grouping of attributes. In next, we applied our proposed Probabilistic Possibilistic Clustering algorithm (PPFCM) for each segment, which produced the number of clusters for each segment. Again, the PPFCM applied on the centroids of the resultant clusters. The corresponding data tuples of the grouped centroids are joined together to attain the final clustered result. Finally, the implementation will be done using JAVA and the performance of the algorithm will be analyzed with benchmark dataset. Our proposed PPFCM performed 5.24 and 6.65% better than possibilistic FCM and probability-clustering algorithm respectively for the mushroom dataset and adult dataset in terms of accuracy. In addition, our proposed PPFCM performed 15.79 and 11.59% better than possibilistic FCM and probability-clustering algorithm respectively for the adult dataset in terms of running time. Moreover, our proposed PPFCM performed 11.57 and 12.80% better than possibilistic FCM and probability-clustering algorithm respectively for the mushroom dataset in terms of running time.

REFERENCES

Adult dataset, 1994. Retrieved from: http://archive.ics.uci.edu/ml/datasets/Adult.

Chen, W.Y., Y. Song, H. Bai, C.J. Lin and E.Y. Chang, 2011. Parallel spectral clustering in distributed systems. IEEE T. Pattern Anal., 33(3): 568-586.

Das, S., A. Abraham and A. Konar, 2008. Automatic clustering using an improved differential evolution algorithm. IEEE T. Syst. Man Cy. A, 38(1): 218-237.

Ester, M., H.P. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), pp: 226-231.

Islam, M.Z. and L. Brankovic, 2004. A framework for privacy preserving classification in data mining. Proceeding of the 2nd Workshop on Australasian Information Security, Data Mining and Web Intelligence and Software Internationalisation, 32: 163-168.

Izakian, H., A. Abraham and V. Snasel, 2009. Fuzzy clustering using hybrid fuzzy C-means and fuzzy particle swarm optimization. Proceeding of World Congress on Nature and Biologically Inspired Computing. IEEE Press, India, pp: 1690-1694.

Jain, Y.K., V.K. Yadav and G. Panday, 2011. An efficient association rule hiding algorithm for privacy preserving data mining. Int. J. Comput. Sci. Eng., 3(7): 2792-2798.

Januzaj, E., H.P. Kriegel and M. Pfeifle, 2004. Scalable density-based distributed clustering. Proceeding of 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp: 231-244.

Ji, J., W. Pang, C. Zhou, X. Han and Z. Wang, 2012. A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. Knowl-Based Syst., 30: 129-135.

Jin, R., A. Goswami and G. Agrawal, 2006. Fast and exact out-of-core and distributed K-means clustering. Knowl. Inf. Syst., 10(1): 17-40.

Kusiak, A. and M. Smith, 2007. Data mining in design of products and production systems. IFAC Annu. Rev. Control, 31(1): 147-156.

Li, T., N. Li, J. Zhang and I. Molloy, 2012. Slicing: A new approach for privacy preserving data publishing. IEEE T. Knowl. Data En., 24(3): 561-574.

Lyigun, C., 2008. Probabilistic Distance Clustering, Proquest, ISBN: 0549980075, 9780549980070.

Mehmed, K., 2003. Data Mining: Concepts, Models, Methods and Algorithms. John Wiley and Sons, Hoboken, N.J.

Mushroom dataset, 1981. Retrieved from: http://archive.ics.uci.edu/ml/datasets/Mushroom.

Ng, R.T. and J. Han, 1994. Efficient and effective clustering methods for spatial data mining. Proceeding of the 20th International Conference on Very Large Data Bases, pp: 144-155.

Osmar, R.Z., 1999. Introduction to Data Mining. In: Principles of Knowledge Discovery in Databases. CMPUT690, University of Alberta, Canada.

Pal, N.R., K. Pal, J.M. Keller and J.C. Bezdek, 2005. A possibilistic fuzzy c-means clustering algorithm. IEEE T. Fuzzy Syst., 13(4): 517-530.

Patel, S., V. Patel and D. Jinwala, 2013. Privacy preserving distributed K-means clustering in malicious model using zero knowledge proof. In: Hota, C. and P.K. Srimani (Eds.), ICDCIT, 2013. LNCS 7753, Springer-Verlag, Berlin, Heidelberg, pp: 420-431.

Roy, B., 2014. Performance analysis of clustering in privacy preserving data mining. Int. J. Comput. Appl. Inform. Technol., 5(2): 35-45.

Sheikholeslami, G., S. Chatterjee and A. Zhang, 1998. WaveCluster: A multi-resolution clustering approach for very large spatial databases. Proceeding of the 24th VLDB Conferences. New York, USA, pp: 428-439.

Wang, W., J. Yang and R. Muntz, 1997. STING: A statistical information grid approach to spatial data mining. Proceeding of the 23rd International Conference on Very Large Data Bases (VLDB), pp: 186-195.

Wehrens, R. and L.M. Buydens, 2004. Model-based clustering for image segmentation and large datasets via sampling. J. Classif., 21: 231-253.

Wu, S. and S. Wang, 2013. Information-theoretic outlier detection for large-scale categorical data. IEEE T. Knowl. Data En., 25(3): 589-602.

Zhang, T., R. Ramakrishnan and M. Livny, 1996a. BIRCH: An efficient data clustering method for very large databases. Proceeding of the ACM SIGMOD International Conference on Management of Data, pp: 103-114.

Zhang, T., R. Ramakrishnan and M.L. Birch, 1996b. An efficient data clustering method for very large databases. Proceeding of the ACM SIGMOD International Conference on Management of Data, pp: 103-114.