**An ensemble approach to outlier detection using some conventional clustering algorithms**

**Abstract**: Outlier detection is an important task in data mining and machine learning. Mining algorithms when applied on datasets having outliers, lead to erroneous conclusion about the data. On the other hand, removing outliers from dataset provides us statistically significant results and we can draw strong conclusion on the data. In this paper, we take an ensemble approach to provide data clustering and outlier detection simultaneously which decides on whether a point is an outlier or not from the combined intelligence of several conventional clustering algorithms. In our model, we make use of clustering algorithms such as K-means, K-means++, and Fuzzy C-means. Our model combines their individual outcome, probabilities to each data point of its belonging to a certain cluster. We then identify outliers based on their assigned probabilities. After removing these data points from the dataset, we get better values of cluster validity indices, ensuring that removal of outliers has resulted in more stringent clustering.

**Keywords:** Outlierdetection, K-means, Fuzzy C-means, K-means++, Ensemble approach

**1. Introduction**

Outliers are unusual values compared to other data points in a dataset. They may arise due to variability in a measurement, experimental errors etc. In general, a dataset overall tends to show a pattern and outliers diverge from that. The most common causes of outliers on a dataset include data entry errors (human errors), measurement errors (instrument errors), intentional (dummy divergent data incorporated to test detection accuracy), natural, sampling errors (extracting and mixing data from wrong or various sources), experimental errors (data extraction or experiment planning/ executing errors), data processing errors (data manipulation or dataset unintended mutations). Common applications of outlier detection techniques include, but not limited to, fraud detection, anomaly detection, unusual medical behaviour etc. If a stolen credit card depicts unusual purchasing behaviour, then property is a clear indication of an outlier, with respect to its native behaviour. Similarly, uncharacteristic symptoms in a patient can be conclusive of a potential health issue.

Some of the primary reasons which make outlier detection a challenging task include difficulty in modelling normal objects and outliers properly, difficulty in handling noise in outlier detection, basis of the detection etc. The demarcation between normal behaviour and outliers is often gray and so classifying certain data objects as outliers can be hard. In different applications, the rate of a normal entity become an outlier varies. For example, in healthcare, small deviation from normal can be suspicious and should be examined. The same is not the case with stock market share values and most other fields. Outlier detection is a well-researched field and hence a varying number of mechanisms have been developed till date to deal with outliers. These mechanisms can be primarily divided into three categories: supervised methods, unsupervised methods and semi-supervised methods. In supervised outlier detection methods, datasets are labelled. Data is primarily classified into two classes. One part of the data belongs to normal class and the remaining part of data belongs to the outlier class. Data objects in the training set are all labelled as either normal or outlier. A classification model is trained on this training data and this trained classification model is then used on unlabeled (or test) data to predict whether those data objects are normal or outliers. This way, supervised outlier detection method essentially reduces the outlier detection problem into a classification problem. In some datasets, we just have the data objects but we do not have any labels associated with the data objects. That is why in this case we cannot train a model to detect outliers. Unsupervised outlier detection makes an implicit assumption, that normal data objects are somewhat clustered. Now, these normal data objects may all belong to the same cluster or they may belong to several clusters. In essence, data objects which are normal follow a pattern far more frequently than outliers. The data objects in a cluster are similar to one another whereas data objects belonging to different clusters are different to one another. In other words, data within the clusters, show a high degree of cohesion and there are high inter-cluster distances.

In this paper, we attempt to find outliers using an ensemble of clustering methods. Specifically, our proposed model assigns probabilities to a data object, of its belonging to a particular cluster. The cluster for which the probability is the highest wins and the data object is assigned to it. Then we shred data objects from clusters in an intelligent way to get rid of points which decrease the cohesiveness of the clusters and label them as outliers. These objects may or may not be outliers. They might be objects which do not have much specific characteristics to be belonged to one cluster over another. But, however getting rid of these data objects give us statistically better results and we get to know the specifics of the individual clusters better. We verify our results by computing the cluster validity indices five popular machine learning datasets such as WDBC, Yeast, Satimage, Ionosphere and Shuttle.

**2. Related work**

Kadam, Pund [1] and Aggarwal [2] studied different approaches to detect outliers. Jiang et al. [3] proposed two initialization methods for the K-modes algorithm which initializes cluster centers carefully so that they are not outliers. Yu et al. [4] designed the OEDP *k* -means algorithm which improved clustering efficiency by removing outliers from the dataset before applying k-means clustering. Aparna and Nair [5] proposed the CHB-K-Means algorithm which used a weighted attribute matrix based mechanism to detect outliers.

Lot of work has been done on outlier analysis. But, only a small part of those studies have performed clustering and outlier detection simultaneously. In this section, we briefly discuss clustering methods with the built-in mechanism of outlier detection. Jiang et al. [6] formulated a two-phase clustering algorithm for outlier detection. In the first phase, a modified version of the K-means algorithm is applied to create new cluster centers for data points which are far away from other clusters. In the second phase, a minimum spanning tree is constructed taking cluster centers obtained from the first phase as vertices of the graph. Clusters in small sub trees are considered as outliers.

Hautamäki et al. [7] designed the Outlier Removal Clustering (ORC) algorithm to identify clusters and outliers from a dataset simultaneously. The ORC algorithm is also a 2-phase process. In the first phase, k-means algorithm is used to cluster the data points. In the second phase, the algorithm iteratively removes the data points that are far away from their cluster centers and marks them as outliers. He et al. [8] introduced a new concept, namely cluster-based local outlier and formulated a quantitative measure, called cluster-based local outlier factor (CBLOF) to identify such outliers in the dataset. Jiang and An [9] also introduced a 2-step algorithm, called Clustering Based Outlier Detection (CBOD). In the first phase, a clustering algorithm is applied to divide a dataset into hyper spheres with almost the same radius. In the second phase, outlier factors for all clusters obtained from the first step are calculated and the clusters are sorted based on their outlier factors. Clusters with high outlier factors are considered outliers.

Zhou et al. [10] proposed a 3-phase modified K-means algorithm to cluster data and detect outliers. In the first phase, fuzzy C-means algorithm is applied on the data. In the second phase, local outliers are identified and removed from the dataset. Hence, the cluster centers need to be recalculated. In the third phase, certain clusters with low inter-cluster separation are merged and global outliers are identified. Zhang et al. [11] introduced a quantitative metric called Local Distance-based Outlier Factor (LDOF) to measure the extent to which a data object can be classified as an outlier in scattered datasets. Jayakumar and Thomas [12] presented an approach to detect outliers based on the Mahalanobis distance. Ahmed and Naser [13] proposed the Outlier Detection and Clustering (ODC) algorithm to detect outliers. The ODC algorithm is a modification of the K-means algorithm. In this algorithm, a data point is marked as an outlier if its distance to its cluster center is at least p times the average distance.

Chawla and Gionis [14] introduced the K-means-algorithm that provided data clustering and outlier detection simultaneously. The K-means-algorithm requires two parameters: K and l, which denote the desired number of clusters and the desired number of top outliers, respectively. Whang et al. [15] formulated the Non-exhaustive Overlapping K-means (NEO-K-means) algorithm, which also identify outliers during the clustering process.

Some of the aforementioned algorithms perform clustering and outlier detection in stages. In these algorithms, a clustering algorithm is used to partition the dataset into clusters and some metric is calculated for the data points based on the clusters to identify outliers. ODC, K-means and NEO-K-means algorithms integrate outlier detection into the clustering process. However, data points that are eliminated as outliers during the iterative process of the ODC algorithm cannot be reconsidered as normal points again when the cluster centers are updated.

In this paper, we propose a 3-phase algorithm which does data clustering and outlier detection simultaneously. In the first phase, we apply a number of conventional clustering algorithms such as K-means, K-means++, Fuzzy C-means to cluster the dataset independently. For soft clustering approaches like Fuzzy C-means, we already have membership scores for each data object to belong in a particular cluster. For hard clustering techniques, we use the distance from a data object to the center of a particular cluster to find its membership score to belong in that cluster. In the second phase, we combine the membership scores obtained in the first phase to get a more reliable result that depends on an ensemble of the said clustering algorithms. In the third phase, we use a threshold to decide whether a data object has membership score high enough to belong to a cluster or it will be flagged as an outlier.

**3. Brief overviews of the clustering algorithms**

Say, we have a dataset with data objects, > 0. The data objects are d-dimensional, i.e., if we denote the data objects by then ∈ d. We need to group the data into clusters. The centers of the clusters are denoted by

**3.1. K-means**

K-means is a hard clustering algorithm which means each data object gets assigned to a single cluster. We initially choose random points from the input as the centers of the clusters. From there on, in each iteration, we perform the following things until we reach a satisfactory value of the objective function:

1. Assign each point to the cluster, the distance to whose center is the least for the point, i.e., assign data object to cluster with center if distance(,) is the least among all distances distance() where .
2. Let, frequency of cluster be denoted by . We update the cluster centers by the following: where if belongs to k-th cluster and otherwise.

**3.2. K-means++**

K-means++ is a variation of the conventional K-means algorithm which differs from the K-means algorithm in terms of initialization of K cluster centers. In this algorithm, the first center is chosen at random from the dataset. For the rest of the initial centers, following principle is applied to choose the centers: select the data point from the dataset whose minimum distance to the already selected centers is the largest.

**3.3. Fuzzy C-means**

Fuzzy C-means is a soft clustering technique. It is based on minimization of the following objective function: m is any real number greater than or equal to 1 and µij is the degree of membership of data point Xi in j-th cluster. Throughout the algorithm, an iterative optimization of the objective function is carried out with updates to the membership function µij and the cluster centers Cj. The updates are given by

and .

The iteration stops when maximum difference in µij value in two subsequent iterations is less than some pre-define constant epsilon. The Fuzzy C-means algorithm converges to a local minimum of the value of the objective function J.

**4. Proposed Methodology**

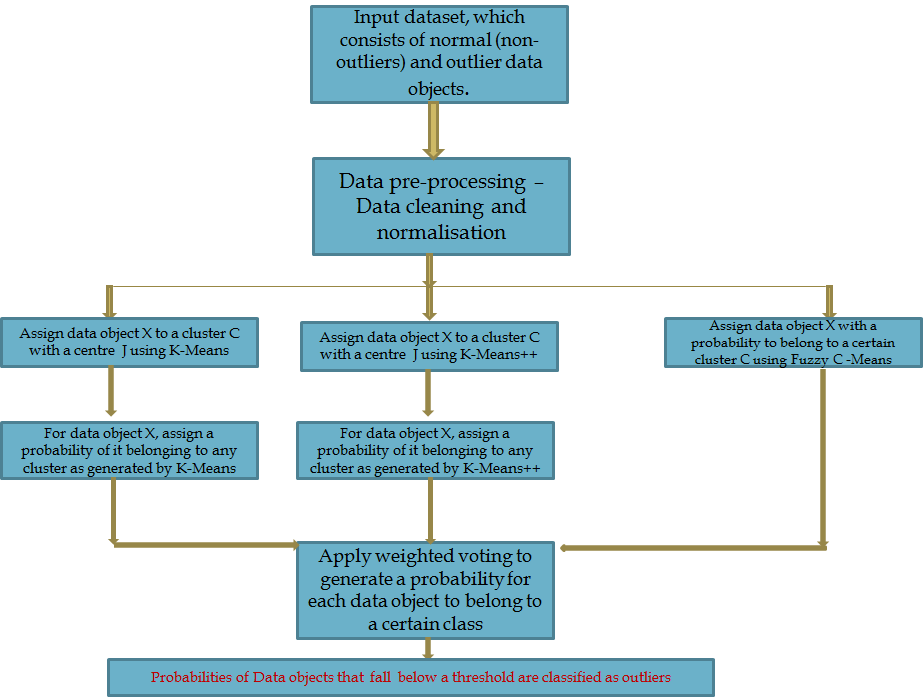


Figure 1 : Overview of our method

In this paper, instead of relying on a single clustering algorithm, we design an ensemble of the said clustering algorithms. Our methodology is illustrated as a flowchart in Figure 1.

None of these clustering algorithms are individually perfect. For example, results from K-means are heavily dependent on the quality of initial center initialization. But with combined intelligence from the ensemble of these methods, we are more likely to rule out the actual outliers. More specifically, we use an ensemble of K-means, K-means++ and Fuzzy C-means. Among these, some are hard clustering techniques, i.e., a particular point gets assigned into a single cluster and some techniques are soft clustering techniques which means instead of assigning a data object strictly to a single cluster, it assigns probabilities to the object to belong in multiple clusters. The cluster for which the probability is the highest; the data object is assumed to be belonging to that cluster. We build a model that combines the results of several algorithms and gives us the final result. Instead of using simple voting, we apply a weighted technique. In a way mentioned below, we get probabilities for a data object to belong to multiple clusters for the hard clustering techniques.

Say, we have a dataset with N data objects, N > 0. The data objects are d-dimensional, i.e., if we denote the data objects by then Xi ∈ Rd. There are K clusters. Their centers are denoted by . Say, after hard clustering, data object belongs to j-th cluster with center . We assign probabilities of to belong to clusters by the following equation:

(1)

Where is the distance between data object Xi and center of k-th cluster Ck.

For soft clustering technique, we already have the probability for data object Xi to belong to the j-th cluster. Probabilities obtained after merging the results from the different clustering techniques can be formulated as follows:

Say, we use an ensemble of m clustering techniques:

, and (2)

gives the combined probability of data object Xi to belong in j-th cluster with center Cj. The data object Xi is assigned to j-th cluster if is maximum among

We infer data object Xi is an outlier if: , (3)

where is a threshold value.

Upon removing these data objects from consideration, improvements are measured by some cluster validity indices such as Dunn, Silhouette, and Calinski-Harabasz.

Given below is an outline of the overall algorithm:

1. Dataset has N data objects and the objects are d-dimensional. The data objects are represented as. There are K clusters and they are represented by their centers, . Perform data pre-processing which includes data cleaning, data normalization so that attributes with higher magnitudes and range do not outweigh attributes with lower value or range.
2. For each of the clustering techniques, assign probabilities to each data object to belong in the K clusters according to eq (1).
3. Combine all these probabilities returned by each clustering technique to obtain a final combined probability vector for each data object according to eq (2).
4. Based on eq (3), decide whether a data object has desired probability to be assigned to the cluster to which it is currently assigned.
5. The data objects that are shred off from the clusters are likely to be outliers.
6. Calculate cohesiveness measuring indices for the clusters before and after elimination of the outliers.

**5. Results and Discussion**

To evaluate the effectiveness of the proposed ensemble based outlier detection model, we have applied the same on some standard machine learning datasets. The proposed algorithm has been evaluated on these given datasets:

1. Wisconsin breast cancer dataset (WDBC) [16]
2. Yeast dataset [17]
3. Satimage dataset [18]
4. Ionosphere dataset [19]
5. Shuttle Dataset [20]

WDBC dataset (Wisconsin breast cancer dataset) has 699 data points and number of clusters is 2. Yeast dataset has 1484 samples. Though the original dataset contains 10 clusters, here for simplicity we assumed that the dataset contains 7 clusters. Ionosphere dataset contains 351 data entries and number of clusters is 2. For Satimage dataset, since it is a clustering task, we only have computed results on satimage.trn data file since no training or testing is required in clustering. It has 4435 data points in the dataset and number of clusters is 7. For Shuttle dataset, we have computed results only on shuttle.tst data file for similar reasons as in Satimage dataset. Here the dataset contains 14500 data entries and number of clusters is 7.

Since for clustering, there is no ground truth data, we determine clustering quality by estimating the values of clustering validity indices. Some of the most widely used clustering indices are:

* Dunn Index
* Silhouette Index
* Calinski-Harabasz Index

Dunn Index: Dunn index is an internal evaluation scheme. Its score is high for clustering results for which there is small variance between members of the cluster and the clusters are sufficiently far apart from each other. Let, m be the number of clusters and the clusters are denoted by . Let, x and y be two d-dimensional data objects assigned to the same cluster Ci. = There can be alternate definitions of . is a measure of intra-cluster variance. Lower the value of this metric, better is the compactness of the cluster Ci.

() is the inter-cluster distance metric between clusters and .

Dunn index value is given by: . For a given assignment of data objects to clusters, higher Dunn index value indicates greater compactness within clusters and higher inter-cluster separation and thus it indicates a better clustering result.

Silhouette Index: Silhouette value measures how close an object is to its own cluster than to other clusters. Silhouette value lies in the range and higher the value, better is the quality of clustering. Higher value means an object is closer to its own cluster compared to other clusters and a lower value means the opposite.

For each data object i, let be average distance of i with all other data residing in the same cluster as i. can be interpreted as how well a data object i is assigned to its own cluster. Let denote lowest average distance of i to all points in any other cluster, which i is not a part of. This cluster is the next best fit for the data object and hence is called the ‘neighbouring cluster’ of data object . For each data object , silhouette value is given by:

.

From the definition we see,. The overall Silhouette value for the dataset is the average of all silhouette values .

Calinski-Harabasz (CH) Index: The mathematical formula for the measure is: , where is the total number of clusters and is the number of observations in the dataset. is the overall intra-cluster variance and is the overall inter-cluster variance. A higher value of the CH index indicates better clustering quality.

Before removing outliers from each dataset, said index values are computed and then after eliminating outliers from the dataset, they are computed again. From the results tabulated in Table 1, we observe that outlier removal results in higher index values for all the three indices, i.e., Dunn, Silhouette, CH. Since, for all these indices, higher value indicates improved clustering quality, we infer that our proposed technique has resulted in increased cohesion within the clusters and higher separation between two different clusters.

We use the proposed model and find values of clustering indices for each of the datasets both before and after performing the procedure.

For datasets where there is not any outlier, the indices values before and after are almost the same. But for datasets with significant fraction of outliers, we observe satisfactory results for clustering indices values. The experimental data is recorded in Table 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Dunn index value** | | **Silhouette index value** | | **CH index value** | |
| **Before analysis** | **After analysis** | **Before**  **Analysis** | **After analysis** | **Before analysis** | **After analysis** |
| WDBC | 1.789819 | 1.832452 | 0.593629 | 0.615671 | 1003.342757 | 1111.701453 |
| Yeast | 1.212317 | 1.762749 | 0.199371 | 0.491729 | 127.834659 | 230.442240 |
| Satimage | 0.447034 | 0.835105 | 0.253721 | 0.250756 | 319.569063 | 2767.223696 |
| IonoSphere | 1.098454 | 1.488427 | 0.310419 | 0.421459 | -7.922495 | 5.890596 |
| Shuttle | 1.257332 | 4.842803 | 0.527632 | 0.775452 | 2282.310773 | 5543.114284 |

Table 1: Experimental outcomes of applying the proposed methodology on 5 standard datasets

As the results shown in Table 1, after eliminating potential outliers from the dataset, the values of the cluster validity indices have improved significantly. The improvements are visualized by bar charts as shown in Figures 2-5.

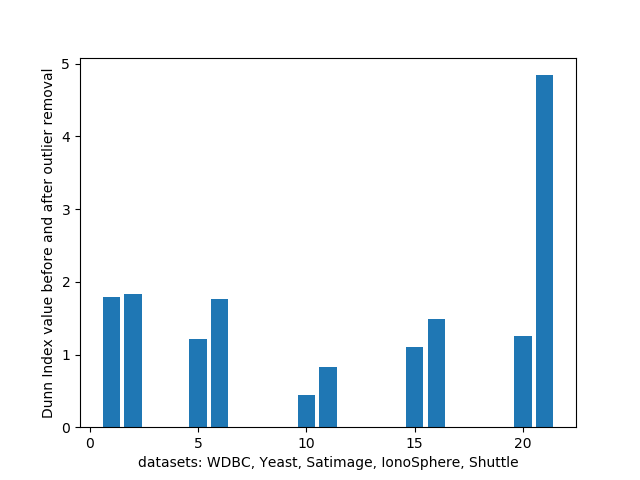


Fig 2: Bar chart showing Dunn index values before and after outlier analysis for WDBC, Yeast, Satimage, Ionosphere and Shuttle datasets.

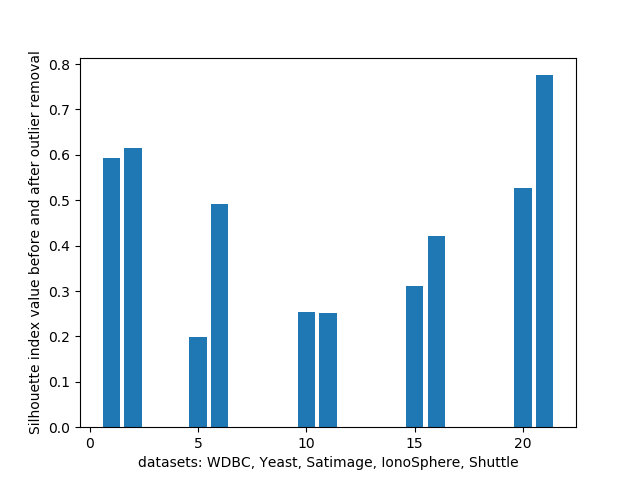


Fig 3: Bar chart showing Silhouette index values before and after outlier analysis for WDBC, Yeast, Satimage, Ionosphere and Shuttle datasets

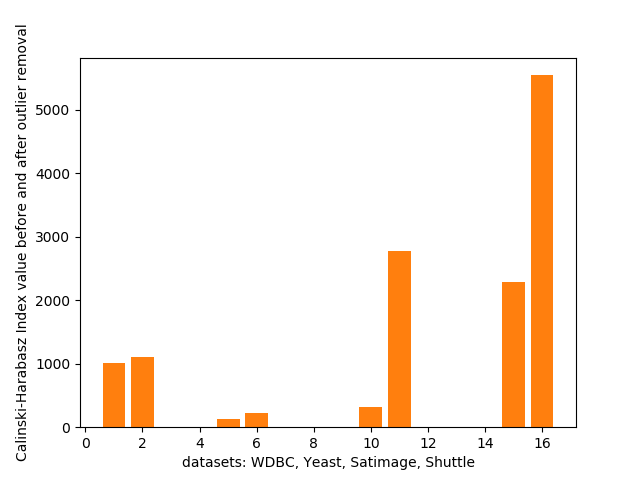


Fig 4: Bar chart showing Calinski-Harabasz index values before and after outlier analysis for WDBC, Yeast, Satimage and Shuttle datasets

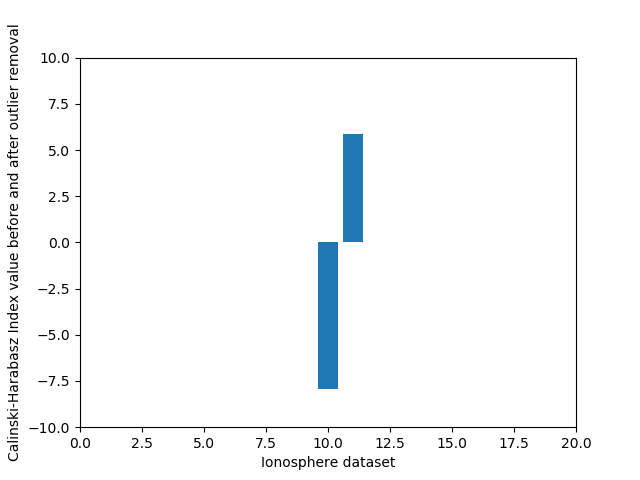


Fig 5: Bar chart showing Calinski-Harabasz index value before and after outlier analysis for Ionosphere dataset.

**6. Conclusion**

In this paper, we propose an ensemble of K-means, K-means++ and Fuzzy C-means clustering algorithms to detect outliers from a dataset. The proposed algorithm yields satisfactory results on some popular datasets. We notice significant improvements in values of cluster validity indices. So, we can safely claim that the proposed model has the ability to detect outliers robustly. In this algorithm, erroneous result yielded by one clustering algorithm gets compensated by results from other clustering algorithms. But, the present algorithm has some limitations. The value of the threshold ϴ is now decided via experimentation and different values of it yield better results for different datasets. Value of ϴ needs to be evaluated in an intelligent manner. This improvement, when incorporated is likely to lead into a better outlier detection mechanism. In the future, we aim to make properly tune our model to enable run-time detection of outliers in a large-scale system.

**References**

1. N.V. Kadam, M.A. Pund, Joint approach for outlier detection, Int. J. Comput. Sci. Appl. 6(2) (2013) 445-448.
2. C.C.Aggarwal, Data Mining: The Textbook, Springer, New York, NY, 2015.
3. F.Jiang, G.Liu, J. Du, Y. Sui, Initialization of k-modes clustering using outlier detection techniques, Inf. Sci. 332 (2016) 167-183.
4. Q. Yu, Y. Luo, C. Chen, X. Ding, Outlier-eliminated k-means clustering algorithm based on differential privacy preservation, Applied Intelligence (2016) 1-13.
5. K. Aparna, M.K. Nair, Computational Intelligence in Data Mining, vol. 2, Springer, pp. 25-35.
6. M. Jiang, S. Tseng, C. Su, Two-phase clustering process for outliers detection, Pattern Recognit. Lett. 22 (6-7) (2001) 691-700.
7. V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, P. Fränti, Improving k-means by outlier removal, in: Proceedings of the 14th Scandinavian Conference on Image Analysis, SCIA’05, 2005, pp. 978–987.
8. Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern Recognit. Lett. 24 (9–10) (2003) 1641–1650.
9. S.-Y. Jiang, Q. An, Clustering-based outlier detection method, in: Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2, 2008, pp. 429–433.
10. Y. Zhou, H. Yu, X. Cai, A novel k-means algorithm for clustering and outlier detection, in: Second International Conference on Future Information Technology and Management Engineering, 2009, pp. 476–480.
11. K. Zhang, M. Hutter, H. Jin, A new local distance-based outlier detection approach for scattered real-world data, in: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD ’09, 2009, pp. 813–822 .
12. G.S.D.S. Jayakumar, B.J. Thomas, A new procedure of clustering based on multivariate outlier detection, J. Data Sci. 11 (2013) 69–84.
13. M. Ahmed, A. Naser, A novel approach for outlier detection and clustering improvement, in: Proceedings of the 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2013, pp. 577–582.
14. S. Chawla, A. Gionis, *k* -means: a unified approach to clustering and outlier detection, SIAM, pp. 189–197.
15. J. Whang, I.S. Dhillon, D. Gleich, Non-exhaustive, overlapping *k* -means, in: SIAM International Conference on Data Mining (SDM), 2015.
16. Street, W. Nick, William H. Wolberg, and Olvi L. Mangasarian. "Nuclear feature extraction for breast tumor diagnosis." Biomedical image processing and biomedical visualization. Vol. 1905. International Society for Optics and Photonics, 1993.
17. Nakai, Kenta, and Minoru Kanehisa. "Expert system for predicting protein localization sites in gram‐negative bacteria." Proteins: Structure, Function, and Bioinformatics 11.2 (1991): 95-110.
18. <https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)> Last Accessed : May 18, 2019
19. Sigillito, Vincent G., et al. "Classification of radar returns from the ionosphere using neural networks." Johns Hopkins APL Technical Digest 10.3 (1989): 262-266.
20. <https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)> Last Accessed : May 18, 2019