



# An iterative approach to unsupervised outlier detection using ensemble method and distance-based data filtering

Bodhan Chakraborty<sup>1</sup> · Agneet Chaterjee<sup>2</sup> · Samir Malakar<sup>3</sup> · Ram Sarkar<sup>2</sup>

Received: 12 August 2020 / Accepted: 28 January 2022 / Published online: 17 February 2022  
© The Author(s) 2022

## Abstract

Outlier or anomaly detection is the process through which datum/data with different properties from the rest of the data is/are identified. Their importance lies in their use in various domains such as fraud detection, network intrusion detection, and spam filtering. In this paper, we introduce a new outlier detection algorithm based on an ensemble method and distance-based data filtering with an iterative approach to detect outliers in unlabeled data. The ensemble method is used to cluster the unlabeled data and to filter out potential isolated outliers from the same by iteratively using a cluster membership threshold until the Dunn index score for clustering is maximized. The distance-based data filtering, on the other hand, removes the potential outlier clusters from the post-clustered data based on a distance threshold using the Euclidean distance measure of each data point from the majority cluster as the filtering factor. The performance of our algorithm is evaluated by applying it to 10 real-world machine learning datasets. Finally, we compare the results of our algorithm to various supervised and unsupervised outlier detection algorithms using Precision@n and F-score evaluation metrics.

**Keywords** Outlier detection · Unsupervised learning · Iterative approach · Ensemble method · Distance-based filtering · Dunn index

## Introduction

Outliers or anomalies are data objects which show different behavior to the rest of the data in a particular dataset. Outliers are generally caused due to errors that occurred during data entry, data measurement, data sampling, data processing along with natural and experimental errors, and many more. Outlier detection is an important application domain

of machine learning and such algorithms are commonly used in fraud detection [1], network industry damage detection [2], healthcare analysis [3], surveillance [4], security [4], intrusion detection [5] and many more.

Detection of outliers is a challenging task as it involves the proper modeling of actual data and outliers. Different flavors of data possess unique challenges. The outlier hypothesis in one domain might not be applicable in another disparate problem. Sometimes, the difference between actual data objects and outliers is minimal, and hence classifying certain abnormalities in data as outliers is quite challenging. The variations on which data objects are classified as outliers vary with the domain of applications [1]. For example, small variations in observed data are neglected in the case of a stock market analysis or fraud detection but in the case of medicinal domains like healthcare such variations cannot be ignored.

The field of outlier detection has been researched extensively in past and various algorithms have been developed to deal with the problem of outlier detection. Such algorithms are generally classified into three common categories namely, supervised method [6], unsupervised method [6], and semi-

---

✉ Samir Malakar  
malakarsamir@gmail.com

Bodhan Chakraborty  
bodhanchakraborty@gmail.com

Agneet Chaterjee  
agneet257@gmail.com

Ram Sarkar  
ramjucse@gmail.com

<sup>1</sup> Institute of Radiophysics and Electronics, University of Calcutta, Kolkata, India

<sup>2</sup> Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

<sup>3</sup> Department of Computer Science, Asutosh College, Kolkata, India

supervised method [4] with each having its own advantages and disadvantages.

In the supervised method of outlier detection, samples in a dataset are labeled and data is classified into two classes: normal and outlier. Data in the training set are labeled as either normal or as an outlier. A classification model is trained on the training set, and then this model is experimented on the test set (unlabeled data) to predict the outliers. Thus supervised outlier detection methods deal with binary classification problems. Such methods can be further sub-divided into statistical methods [7], decision tree based [8], proximity based approaches [9], etc. The work done in this domain comprises the application of particle swarm optimization algorithms to detect intrusions such as PROBE, R2L, U2R, and DoS. This method was applied for the detection of fraudulent credit cards. In the other work [9], a DBSCAN based outlier detection technique is applied, in combination with Luhn's and Bayes' theorem to develop a probability metric that estimates the validity of the credit card.

In cases where labeled data are not available, researchers use unsupervised outlier detection techniques to predict the outliers. In the unsupervised outlier detection approach, the data are assumed to be following a certain pattern that is normal data objects form clusters with high cluster density (i.e., low intra-cluster distance) and high inter-cluster distance. Outliers are detected based on their affinity of belonging to a data cluster. Unsupervised outlier detection methods have been used in various domains such as intrusion detection [10] and sensor networks [11] etc. Zhang et al. [10] develop a random forest based algorithm to identify anomalous patterns in network data. They specifically look at identifying data points that deviate significantly in the same network service and points which belong to other network services. On the other hand, in the work [11], the authors primarily focused on creating an aggregate tree by disseminating beacon information from a sink node and recursively calculating the outlier score for each descending child node.

Semi-supervised learning falls in between supervised and unsupervised methods. The semi-supervised method uses a small amount of labeled data with a very large amount of unlabeled data. In the semi-supervised method of outlier detection, the similar data are first clustered using an unsupervised learning algorithm and then the unlabeled data are labeled based on the existing labeled data into normal and outlier classes. A few of the semi-supervised techniques are followed in the works [12, 13]. Dasgupta and Majumder [12] described a nature inspired censoring process of receptors and apply it to multidimensional personnel data. It is a two-step process wherein given a set of self-strings  $S$  and a matching threshold  $r$ , the algorithm's first phase finds the total number of unmatched strings for the defined self ( $S$ ); the second phase selects some of them to build a varied group of detectors for monitoring data patterns.

As elucidated above, both supervised and semi-supervised based outlier detections, depend upon labeling of data, which can be very expensive in certain domains. Furthermore, only unsupervised approaches can classify unknown data into the right bucket of classifiers or not. Our work aims to propose a generic algorithm, which can coherently perform outlier detection on a myriad of data points. This is the major reason for prevailing with unsupervised learning techniques as the underpinning for our research.

Conventional clustering algorithms are insufficient for outlier detection as none of these clustering algorithms provides high accuracy in detecting the actual outliers. Current works in this domain have a multitude of shortcomings [14] that need to be alleviated. To be specific, numerous outlier detection methods (e.g., [15, 16]) require different parameters that are mostly manually tuned. These parameters, if not tuned optimally, are very sensitive to the end outcome. On the contrary, the method we propose here needs only an initial number of clusters and the threshold value as manual parameters. Furthermore, unlike our method, deep learning based outlier detections require a huge number of training samples and have a high time complexity which might not be feasible in terms of real-world data. Furthermore, many existing algorithms do not discount for sparsely arranged or differentiate between the subtle boundaries between two groups. Noting this and the differentiation between noise and outliers, our method intelligently reconsiders all possible candidates, and makes an informed decision. Finally, developing a generalized outlier detection has always been challenging, however, through our application, we go on to validate our results, irrespective of the domains.

Hence, the use of an ensemble method that combines the results obtained from multiple clustering algorithms instead of relying on a single algorithm proves to be a better solution for outlier detection purposes. Furthermore, a distance-based data filtering method has been used as the final layer to further increase the outlier detecting capabilities of the ensemble. Three commonly used clustering techniques: K-means, K-means++, and Fuzzy C-means have been used as the base for our algorithm. All the datasets used in our experiment have been pre-processed, cleaned, and normalized. The ensemble method is used to cluster the unlabeled data and to detect potential isolated outliers based on the cumulative degree of belongingness generated by combining the degree of belongingness scores obtained from each clustering method. An iterative process is designed to maximize the value of the Dunn index (commonly used evaluation metric for clustering methods) which gives us the optimum point. In each step of the iteration, the possible outliers are eliminated from the dataset and the inliers which had been labeled as an outlier in the previous iteration are added to the new dataset based on a cluster membership threshold value. The resultant post-clustered dataset is screened based on a distance-based

threshold relative to the center of the majority cluster and further possible outlier points are eliminated. The top  $N$  outlier points are chosen based on distance from the majority cluster center. The results have been computed and compared with other outlier detection algorithms for 10 popular machine learning datasets of different sizes and dimensionalities. The contributions of our work are as follows:

- An iterative weighted ensemble of two hard clustering algorithms and one soft clustering algorithm is explored, eliminating bias and introducing agility in terms of a robust search space.
- A Dunn index based thresholding criterion is developed, which ensures that the outlier search is always progressive.
- An intelligent learning step is proposed where previously deemed outliers are also put in contention of being inliers, which removes probabilistic bias present initially and eliminates class imbalance in terms of the number of outliers.
- The results are quantified on 10 openly available datasets and are compared against 12 state-of-the-art algorithms with good results.

## Related work

Several solutions have been proposed in the field of outlier detection spanning multiple domains. It has been helpful in intrusion detection, fraud detection, time-series monitoring, and many more. Zhang et al. [10] use the Random Forest algorithm to detect anomalies by building patterns of network intrusion over traffic data. Similarly, fraud detection is dealt with in the work [17], where the authors develop a replicator neural network and apply it to discover oddities across clusters. The work mentioned in [18] tackles several problems which are prevalent in outlier detection of both univariate and multivariate financial time series. It looks at calculating the projections with maximal kurtosis under a finite mixture model which enables separation of outliers from the bulk of the data. Zhang et al. [19] introduce a new factor, Local-Distance-based Outlier Factor (LDOF) which calculates the degrees of outlier-ness of an object in real-world datasets. LDOF is calculated based on the relative location of an object to its neighbors and the false-detection probabilities. The algorithm [20] calculates outliers at a nearly linear time when the dimensionality of the dataset is high. It tracks the nearest neighbors of a point and removes the point from a potential outlier list, using a certain threshold value. At every iteration, the threshold value increases with the pruning efficiency. However, the major disadvantage of this algorithm is that when there are not many outliers in the system, the running time of the algorithm becomes quadratic. To solve this, a two-step algorithm is developed in [21], which tries

to solve the bottleneck of high-dimensional data by performing a pre-processing step that allows for fast determination of approximate nearest neighbors. K-means is initially used by Jiang et al. [22] to solve the problem of outlier detection by following a two-phase approach, where they first modify the heuristic of K-means to separate out data points into a new cluster, which are far away from each other, followed by generating a Minimal Spanning Tree (MST) to delete the longest (farthest) edge.

More recently, K-means coupled with a Genetic Algorithm (GA) has been used to solve the said problem. For example, in [23], GA has been used in outlier detection of sparsely populated datasets, where it is used to insert large amounts of relevant data in sparse regions which is then used by the K-means algorithm to solve the interpolation problem. Similarly, Triangle Area-based nearest neighbors are introduced in [24] which works on top of the centroids generated by the K-means algorithm, uses triangular area on each centroid to generate several sets of data, and then uses the  $k$ -nearest neighbor ( $k$ -NN) algorithm for further analysis. The problem of deriving  $K$ , the number of clusters, is done away in [25] where the authors generate a series of local proximity graphs based on a uniform sampling strategy, followed by performing Markovian random walks to detect anomalies across these graphs. Similarly, algorithms have been developed which are built on top of K-means++ and Fuzzy C-means. Feng et al. [26] develop an approximation algorithm for general metric space which uses the K-means++ to sample points from the space and then use them to select centers from them, whereas Fuzzy C-means has been used in [27] to generate clustering extreme classes (highest positive and negative), which are then trained by a Support Vector Machine (SVM) to produce the final classification results. The above works mostly suffer from sensitivity to the initial centroid formation and may not be useful on large scale data.

Ensemble techniques have also been worked upon to solve the problem of outlier detection. For example, Aggarwal in [28] talks about the categorization of outlier ensemble algorithms into sequential or independent ensembles and model-centered or data-centered ensembles; which enables in identifying the metrics behind the ensemble approach. K-means has been used as an ensemble in [29], where the algorithm is used multiple times in tandem to obtain a weighted graph structure equivalent to the averaged co-membership matrix. Neural networks coupled with data and edge sampling have also been used in [30] as an ensemble outlier detection approach, achieved by randomly varying the connectivity architecture of the auto-encoder, obtaining significantly better performance. Most ensemble methods suffer from the problem of either normalization or combination. Given a set of outlier patterns, the problem arises in comparing the different results, since each component of the

ensemble might represent its results in a different reference. Similarly, given a normalized set of outlier scores, the technique of combining them together also becomes challenging. For example, if an ensemble has the application of a fuzzy-based and a hard clustering algorithm, combining them so that they can be mapped onto the same proportion is often a tricky problem.

## Present work

Our proposed algorithm focuses on the detection of outliers that may be isolated, clustered, or a combination of both. To differentiate between those outliers, we divide our algorithm into two stages. The first stage uses the clustering ensemble which iteratively clusters as well as removes the potential isolated outliers if any from the unlabeled data based on a preset cluster membership threshold value. The second stage uses the distance-based data filtering following a preset distance threshold to filter out the potential outlier clusters if any from the filtered results (post-clustered data) we obtained from the first stage of our algorithm. In this section, we first explain all prerequisites of the proposed method, and then finally we describe the proposed technique.

## Clustering algorithms used

As previously mentioned, we use three conventional clustering algorithms: K-means, K-means++, Fuzzy C-means out of which two are hard clustering algorithms and the other follows the soft clustering paradigm. The degree of belongingness scores is not generated by hard clustering algorithms as data objects can only belong to a single cluster (class). Hence, we generate the degree of belongingness scores using the formula described in section “[Degree of belongingness score generation for clustering algorithms](#)”. In the case of the soft clustering algorithm, the data objects belong to multiple clusters (classes) with varying degrees of belongingness (i.e., the fuzzy membership values) of the object belonging to each cluster and thus manual generation of the degree of belongingness scores are not required. Below we describe the three algorithms which form our clustering based ensemble.

### K-means

K-means [31, 32] is a hard clustering algorithm. It uses an iterative process that partitions data into  $K$  non-overlapping clusters whose centroids are chosen randomly at an initial step from the dataset. The K-means algorithm tends to maximize the inter-cluster distance and minimize the intra-cluster distance. A data point is assigned to a cluster that has its centroid at a minimum distance from that data point. The iterations are performed till the algorithm converges (i.e.,

objective function reaches a satisfactory value). The steps of the K-means algorithm are given below:

1. Specify the number of clusters  $K$  for a  $d$ -dimensional dataset.
2. Initialize centroids by first shuffling the given dataset and then randomly selecting  $K$  points for the cluster centroids from the shuffled dataset without replacement.
3. Keep iterating until the algorithm converges.
  - 3.1 Assign each data object  $x_i$  to the cluster whose centroid  $c_j$  is at a minimum distance,  $d_{ij} = \min(\|x_i - c_j\|)$  from the data object  $x_i$ , where  $j = 1, 2, 3, 4, \dots, K$ .  $\|\cdot\|$  is a norm function.
  - 3.2 Let the number of initial data objects in the  $j$ th cluster be  $N_j$  with its centroid denoted by  $c_j$ . Then update the cluster centers using the formula:

$$c_j = \frac{1}{N_j} \sum x_i \mu_{ij}. \quad (1)$$

In Eq. (1),  $\mu_{ij} = 1$  if  $x_i$  belongs to the  $j$ th cluster else  $\mu_{ij} = 0$ .

### K-means++

Like K-means, K-means++ [33] is also a hard clustering algorithm. It is the modified version of the K-means algorithm which has faster convergence than the conventional K-means algorithm and provides a better clustering of unlabeled data with high inter-cluster distances. It uses a different initial cluster centroid assignment method. The very first cluster centroid is chosen at random from the dataset. Then the remaining  $(K - 1)$  cluster centroids are selected based on the principle that a data point from the dataset whose minimum distance to the already selected cluster centroids is the largest is chosen as the new cluster centroid.

### Fuzzy C-means

Fuzzy C-means [34] is a soft clustering algorithm. Fuzzy C-means allows data to belong to more than one cluster. It is based on the minimization of the objective function,

$$J_m = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m x_i - c_j^2, \quad 1 \leq m < \infty. \quad (2)$$

Here in Eq. (2),  $m$  is any real number greater than 1,  $\mu_{ij}$  is the degree of membership of  $x_i$  in the cluster  $j$ ,  $x_i$  is the  $i$ th datum of  $d$ -dimensional dataset,  $c_j$  is the centroid of the  $j$ th cluster in  $d$ -dimensional space.  $N$  is the number of objects in the dataset and  $C$  is the number of cluster centers and  $\|\cdot\|$  is the norm function. Fuzzy partitioning is carried

out through an iterative optimization of the objective function (see Eq. (2)), with the updated membership value (say,  $\mu_{ij}$ ) and the cluster centers (say,  $c_j$ ) defined in Eqs. (3) and (4), respectively:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}, \tag{3}$$

$$c_j = \frac{\sum_{i=1}^N \mu_{ij}^m \cdot x_i}{\sum_{i=1}^N \mu_{ij}^m}. \tag{4}$$

This iteration will stop when  $\max_{ij} \left\{ \left| \mu_{ij}^{k+1} - \mu_{ij}^k \right| \right\} < \varepsilon$ , where  $\varepsilon$  is an iteration termination criterion between 0 and 1, and  $k$  is the iteration count. This procedure converges to a local minimum of the objective function  $J_m$ .

The algorithm can be described by the following steps:

1. Initialize  $U = [\mu_{ij}]$  matrix,  $U^0$  for  $k = 0$ .
2. At  $k$ th-step: calculate the centroid vectors  $C^k = [c_j]$ , where  $c_j$  is defined using Eq. (4).
3. Update  $U^k$  to  $U^{k+1}$ , where each element  $\mu_{ij}$  in the  $U$  matrix is defined using Eq. (3).
4. If  $\|U^{k+1} - U^k\| < \varepsilon$  then terminate the process otherwise continue the process to  $(k + 1)$ th step.

### Degree of belongingness score generation for clustering algorithms

The degree of belongingness is the scoring scheme we use to assign cluster membership to unlabeled data objects. The hard clustering algorithms in use assign the data objects to a single cluster hence the degree of belongingness is not generated by those algorithms. In the case of soft clustering algorithms, the fuzzy membership values (which are equivalent to the degree of belongingness scores) form an integral part of the algorithm and are crucial for the clustering of unlabeled data. For soft clustering algorithms, a data object belongs to a cluster for which it holds the highest membership value. So, the degree of belongingness scores for a soft clustering algorithm can be represented by its membership values. The degree of belongingness scores for the hard clustering algorithms is defined using the formula:

$$P_{ij} = 1.0 - \left( \frac{\|x_i - c_j\|}{\sum_k \|x_i - c_k\|} \right). \tag{5}$$

Here,  $j = 1, 2, \dots, K$  and  $P_{ij}$  represents the degree of belongingness of data object  $x_i$  of belonging to  $j$ th cluster with centroid  $c_j$ . From Eq. (5) we observe that degree of belongingness score  $P_{ij}$  can have a maximum value of 1.0 and a minimum value of 0.0. So, we set the scores according to the Euclidean distance of a data point from a cluster

centroid. Thus, according to the ratio  $\frac{\|x_i - c_j\|}{\sum_k \|x_i - c_k\|}$  for larger  $\|x_i - c_j\|$  values the overall value will be smaller and hence  $P_{ij}$  will have a smaller degree of belongingness score for larger distance.

The degree of belongingness scores (equivalent to the fuzzy membership values) for the soft clustering algorithms are generated using the formula:

$$P_{ij} = \mu_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}. \tag{6}$$

In Eq. (6),  $j = 1, 2, \dots, K$  and  $P_{ij}$  represents the degree of belongingness of data object  $x_i$  of belonging to  $j$ th cluster with centroid  $c_j$  and  $1 < m < \infty$  ( $m$  represents the amount of fuzziness in our dataset).

### Weighted method of generating the cumulative degree of belongingness scores

In “Degree of belongingness score generation for clustering algorithms” section, we discuss how the degree of belongingness can be generated for the hard clustering algorithms. To combine the degree of belongingness scores we generated for each clustering technique in the ensemble, we use a weighted technique which we represent using the formula provided in Eq. (7):

$$P_{ij} = \frac{\prod_y P_{ij}(y)}{\sum_j \prod_y P_{ij}(y)}. \tag{7}$$

In Eq. (7),  $y = 1, 2, \dots, m$  and  $j = 1, 2, \dots, K$ .  $P_{ij}$  is generated for an ensemble of  $m$  clustering techniques which is in our case is 3.  $P_{ij}$  is the combined degree of belongingness of the data object  $x_i$  of belonging to  $j$ th cluster with centroid  $c_j$ .

The data object  $x_i$  is assigned to a cluster for which it has the maximum  $P_{ij}$ . Hence, we classify the point  $x_i$  as an outlier if:

$$\max\{P_{ij}\} < P_{th}. \tag{8}$$

In Eq. (8),  $P_{th}$  is the cluster membership threshold. Only those data objects with a combined degree of belongingness scores greater than the cluster membership threshold are considered as inliers by our clustering based ensemble method.

### Dunn index maximization for convergence of iteration

Different clustering validity indices are used to measure the quality of a clustering algorithm. Dunn index [35, 36] scores are one of the widely used clustering validity indices. It is an

internal evaluation scheme. The higher Dunn index indicates greater compactness within clusters and higher inter-cluster separation. Dunn index values are maximized over an iterative process to locate the optimum point (the best quality of clustering). Dunn index values are calculated for each iterative step using Eq. (9):

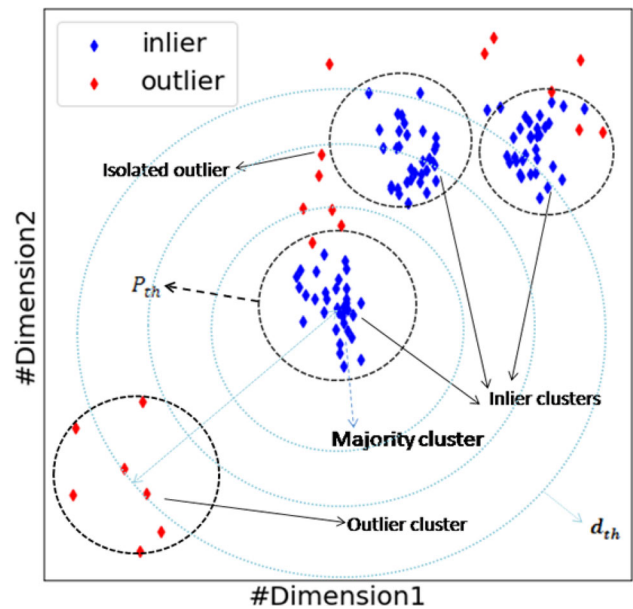
$$\text{Dunn}_K^t = \frac{\min_{i \leq i < j \leq K} \{d(c_i, c_j)\}}{\max_{1 \leq p \leq K} \Delta_p} \quad (9)$$

In Eq. (9),  $t$  is the iterative step count,  $d(c_i, c_j)$  is the inter-cluster distance measure between cluster centroids  $c_i$  and  $c_j$  and  $\Delta_p$  is the intra-cluster distance measure for the  $p$ th cluster.

In our algorithm, we use an iterative process to maximize the Dunn index values (maximize the quality of cluster formation). In an iterative step, a selective number of data points with a degree of belongingness less than the threshold level ( $P_{th}$ ) are eliminated from the dataset. Then in the next step, the ensemble of the clustering techniques is again applied to the resultant dataset and the degree of belongingness scores for both the previously eliminated data as well as data points in the current dataset are computed. Precisely, we have not considered any data point as an absolute outlier point at any step of the iteration. We consider a data point as an outlier at the end of the iterative process. Based on the new degree of belongingness scores, we perform three operations:

1. Some of the previously eliminated data points which have been labeled as outliers in the  $t$ th iteration is again included in the dataset at  $(t + 1)$ th iteration based on the assumption that  $P_{oij}^{t+1} > P_{th}$  i.e., if the data point satisfies the condition to be inlier. Here  $P_{oij}^{t+1}$  is the updated degree of belongingness scores for the previously eliminated data point in the  $(t + 1)$ th iterative step, for  $i = 1, 2, \dots, M$  (where  $M$  is the number of previously eliminated data points),  $j = 1, 2, \dots, K$  (where  $K$  is the number of clusters).
2. Data points from the new dataset are eliminated and labeled as outliers in the  $(t + 1)$ -th iteration based on the assumption that  $P_{nij}^{t+1} < P_{th}$ . Here  $P_{nij}^{t+1}$  is the updated degree of belongingness scores for the new data points in the  $(t + 1)$ -th iterative step, for  $i = 1, 2, \dots, N'$  (where  $N'$  is the number of data objects in the new dataset),  $j = 1, 2, \dots, K$  ( $K$  is the number of clusters).
3. We update the current Dunn Index value (i.e., Dunn Index for the  $(t + 1)$ th iterative step) and continue with the next iterative step if  $\text{Dunn}_K^{t+1} > \text{Dunn}_K^t$  where,  $\text{Dunn}_K^t$  is the Dunn Index for the  $t$ th iterative step, else we terminate the iteration.

The resultant Dunn index value after the termination of the iterative process is the maximized Dunn index value (i.e.,



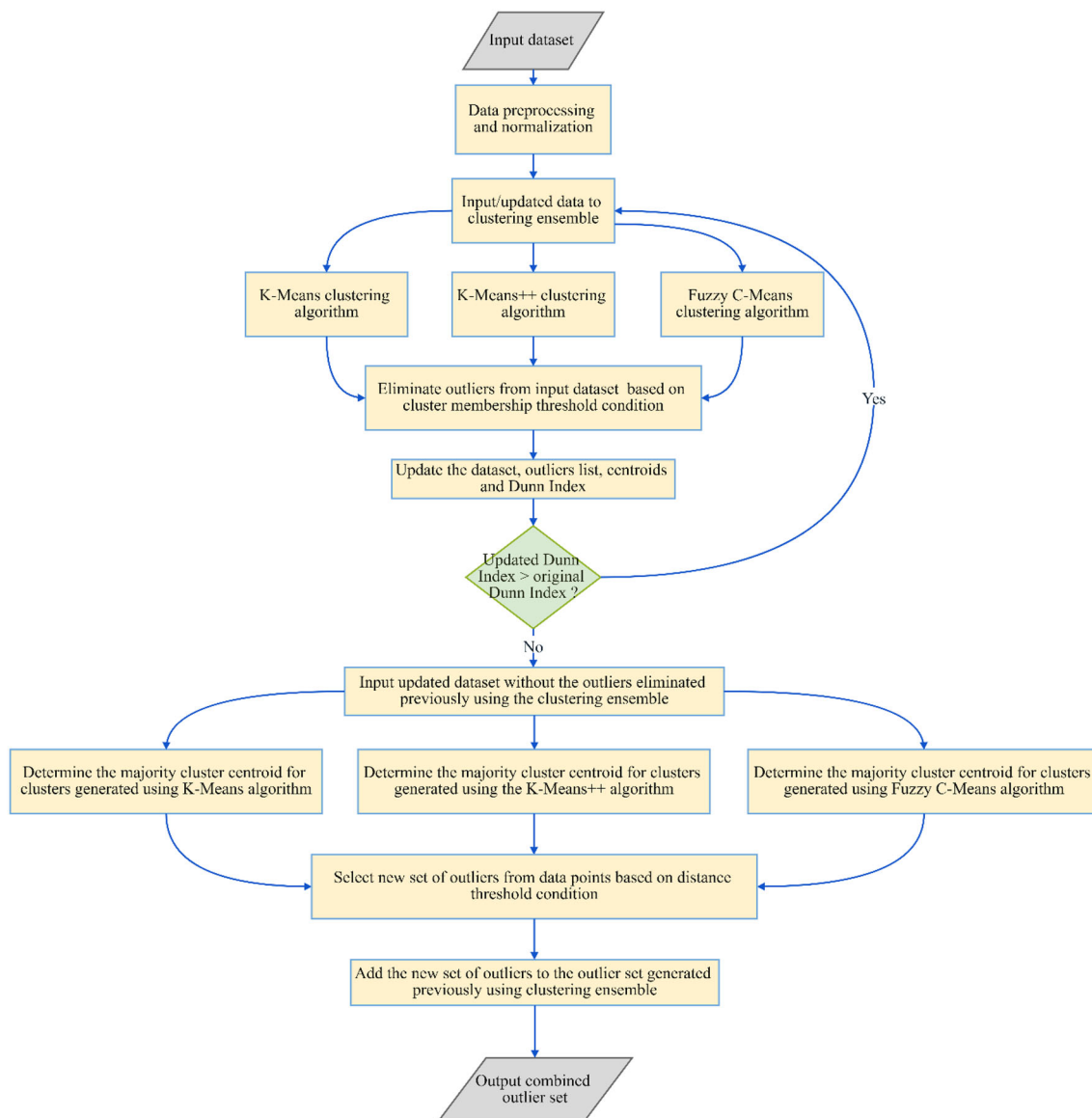
**Fig. 1** Pictorial description of the proposed method (iterative ensemble method with distance-based data filtering). The circles with black colored dashed lines show the elimination process of outliers based on a cluster membership threshold ( $P_{th}$ ), in which the points outside the dashed circles are eliminated. The concentric blue circles show the process of elimination of outliers in outlier clusters based on a distance threshold ( $d_{th}$ ) from the majority cluster center

Dunn index that provides the best quality of clustering) which provides the optimal point of operation of our algorithm. The data points which are eliminated in the final step of the iterative process are labeled as actual outliers.

### Distance-based filtering model

Outliers though small in number when compared with the number of actual data points in the dataset may share common traits with other outliers and may form clusters of their own. Let us assume that the outliers form the number of clusters with relatively smaller cluster densities than that of actual ones, but with large inter-cluster distance and small intra-cluster distance. In this scenario, we cannot eliminate those clusters using the iterative ensemble of clustering techniques due to the small intra-cluster distance between outlier points and hence with the degree of belongingness ( $P_{ij}$ ) greater than that of the fixed threshold ( $P_{th}$ ). To eliminate those clusters, we use a distance-based filtering model which performs the following operations on the post-clustered dataset formed using the iterative ensemble method:

1. Fix a threshold distance ( $d_{th}$ ) for the filter.
2. Determine the majority cluster (the cluster with the highest cluster density value) of the post-clustered dataset for



**Fig. 2** Overview of our proposed method (iterative ensemble method with distance-based data filtering) for outlier detection

each of the clustering techniques (K-means, K-means++, and Fuzzy C-means) used in the ensemble.

- For each of the three clustering techniques compute the distance of each object  $x_i$  in the dataset from the centroid of the majority cluster  $C_k$  (centroid for  $k$ th clustering method). Label the object  $x_i$  as an outlier if  $\min_k \|x_i - C_k\| > d_{th}$  where  $i = 1, 2, \dots, N''$  and  $N''$  represents the size of the post-clustered dataset and  $k = 1, 2, 3$  denotes the number of clustering techniques used in the ensemble. To be specific, for each of the clustering techniques, if an object  $x_i$  is labeled as an outlier, then the point is labeled as an actual outlier despite its belongingness in a formed cluster by present ensemble-based clustering.

### Iterative ensemble method with distance-based data filtering

In the previous sections, we discussed in detail the different basics and working principles of our algorithm. In this section, we combine the various steps mentioned in the previous sections and present the overall algorithm. A pictorial description of the proposed method has been shown in Fig. 1 while in Fig. 2, the methodology of the proposed method is shown using a flowchart.

Given a dataset  $X = \{x_1, x_2, \dots, x_n\}$ , the number of clusters (say,  $K$ ), and fixing a cluster membership and distance threshold as  $P_{th}$  and  $d_{th}$  respectively. The steps of the algorithm are as follows:

1. Compute clusters for the given dataset using each of the mentioned three clustering techniques.
2. Determine the cluster centroids and degree of belongingness of each of the data point  $x_i$  in the dataset for each of the clustering techniques used. For hard clustering algorithms (i.e., K-means and K-means++) assign the degree of belongingness scores to data points based on the formula defined in Eq. (5). For the soft clustering algorithm use the fuzzy membership values as the degree of belongingness scores for the data points based on the formula defined in Eq. (6).
3. Start the iterative process and compute  $Dunn_K^t$  where  $t = 0$  for the initially computed clusters.
4. Generate the cumulative degree of belongingness score for the cluster ensemble using the formula defined in Eq. (7). Eliminate points from the given dataset and label them as possible outliers using Eq. (8) based on the maximum cumulative degree of belongingness score of each data point  $x_i$  of  $X$ .
5. Compute the clusters and centers for each of the clustering techniques for post elimination. Find the updated Dunn index value ( $Dunn_K^t$ ) for updated clusters in the  $t$ th iterative step.
6. In  $(t + 1)$ th iterative step, check if any of the data points in the updated dataset have a degree of belongingness  $P_{n_{ij}}^{t+1} < P_{th}$  for which those points are eliminated from the dataset. Also, check if any of the previously eliminated possible outliers have a degree of belongingness  $P_{o_{ij}}^{t+1} > P_{th}$  (i.e., satisfies the condition to be inlier) for which those points are included in the updated dataset.
7. Check if updated Dunn Index value (i.e., Dunn Index for the  $(t + 1)$ th iterative step) is greater than original Dunn Index value (i.e., Dunn Index for the  $t$ -th iterative step)  $Dunn_K^{t+1} - Dunn_K^t > 0$ . If true then continue with the next iteration step else we terminate the iteration process
8. After termination of the iteration determines the final majority clusters for each of the 3 clustering techniques. Based on the distance threshold  $d_{th}$  select points  $x_i$  from the dataset for which  $\min_k \|x_i - C_k\| > d_{th}$  where  $k = 1, 2, 3$  (number of clustering techniques used),  $C_k$  is the majority cluster centroid for the  $k$ th clustering method. If a point  $x_i$  is labeled as an outlier by all three of the clustering methods, then select that point as an outlier and add that point to the list of previously labeled outliers (outliers labeled using the ensemble).

## Results and discussion

To assess the performance of our proposed algorithm, we evaluate it on 10 machine learning datasets with different sizes and dimensions and each having a varying number of

outliers. Our results have been compared to those obtained from 12 other outlier detection algorithms which demonstrate that our proposed method shows superior or comparable performance on most of the datasets to those existing outlier detection algorithms. In this section, first, we describe different requirements for the experiment and then mention the obtained results.

### Database description

The datasets on which our proposed method has been evaluated are shown in Table 1. From Table 1, we observe that the Pima dataset has the largest number of samples (510) and the Parkinson dataset is the smallest sized dataset (50). The Ionosphere dataset has the largest dimension (32), whereas the Glass dataset has the smallest dimension (7). In terms of outlier count Ionosphere, the dataset has the largest number of outliers at 126 (i.e., 35.90% of the entire dataset) with the Pima dataset is having the smallest outlier count at 10 (1.96% of the entire dataset).

### Evaluation metrics

We evaluate the performance of our algorithm based on the evaluation metrics described below.

**F-score:** The F-score [37, 38], also called the F1-score or the F-measure reflects upon a test's accuracy. It is defined as the harmonic mean of the precision and recall values. This score is calculated according to:

$$\begin{aligned} \text{F-score} &= \left( \frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} \\ &= 2 \times \left( \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \right). \end{aligned} \quad (10)$$

With both precision and recall considered for the calculation of the F1-score, it is useful in providing a realistic measure of a test's performance by balancing both precision and recall. A F-score value ranges from 0 to 1.

**Precision@n ( $P@n$ ):** For a classification system  $P@n$  [39, 40] is the proportion of relevant output at the top- $n$  selected outputs. For example, if our algorithm returns  $N$  number of ranked data points as outliers and out of those  $N$  points if we use only the top- $n$  ranked data points as outlier points of which  $r$  points are true outliers then we calculate  $P@n$  of that algorithm as:

$$P@n = \frac{r}{n}. \quad (11)$$



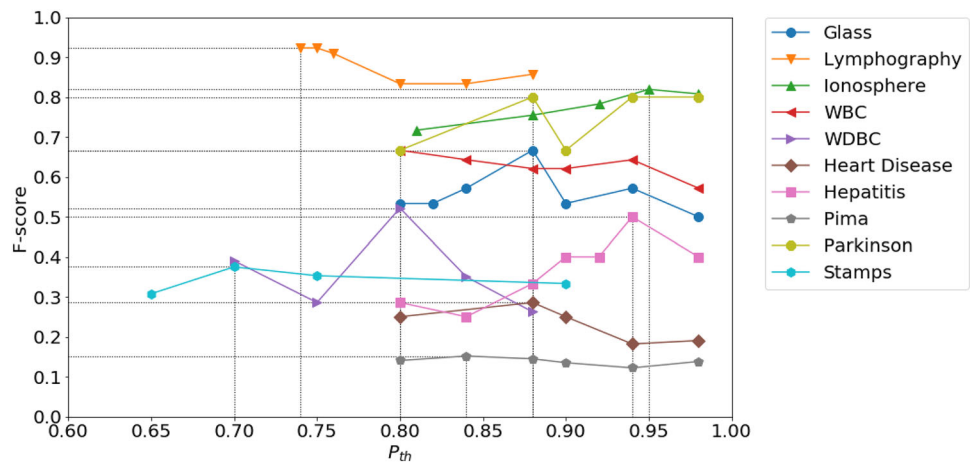
**Table 1** The datasets used to evaluate the proposed outlier detection method and their properties

Dataset	# samples	Dimension	# outliers	% of outliers
Glass	214	7	9	4.21
Lymphography	148	18	6	4.05
Ionosphere	351	32	126	35.90
WBC	223	9	10	4.48
WDBC	367	30	10	2.72
Heart disease	153	13	3	1.96
Hepatitis	70	19	3	4.29
Pima	510	8	10	1.96
Parkinson	50	22	2	4.00
Stamps	315	9	6	1.90

**Table 2** Results for the present iterative ensemble method with distance-based filtering. Here, the optimal parameter values are mentioned that we obtain after several experiments

Dataset	F-score	Best parameters		
		$K$	$d_{th}$	$P_{th}$
Glass	0.66667	3	0.66	0.88
Lymphography	0.92308	2	0.63	0.74
Ionosphere	0.81928	7	0.72	0.95
WBC	0.66667	2	0.36	0.80
WDBC	0.52174	2	0.39	0.80
Heart disease	0.28572	15	0.59	0.88
Hepatitis	0.50000	12	0.87	0.94
Pima	0.15151	14	0.70	0.84
Parkinson	0.80000	10	0.69	0.98
Stamps	0.37500	6	0.49	0.70

**Fig. 3** Variation of F-score with the parameter value ( $P_{th}$ ) for the present iterative ensemble method

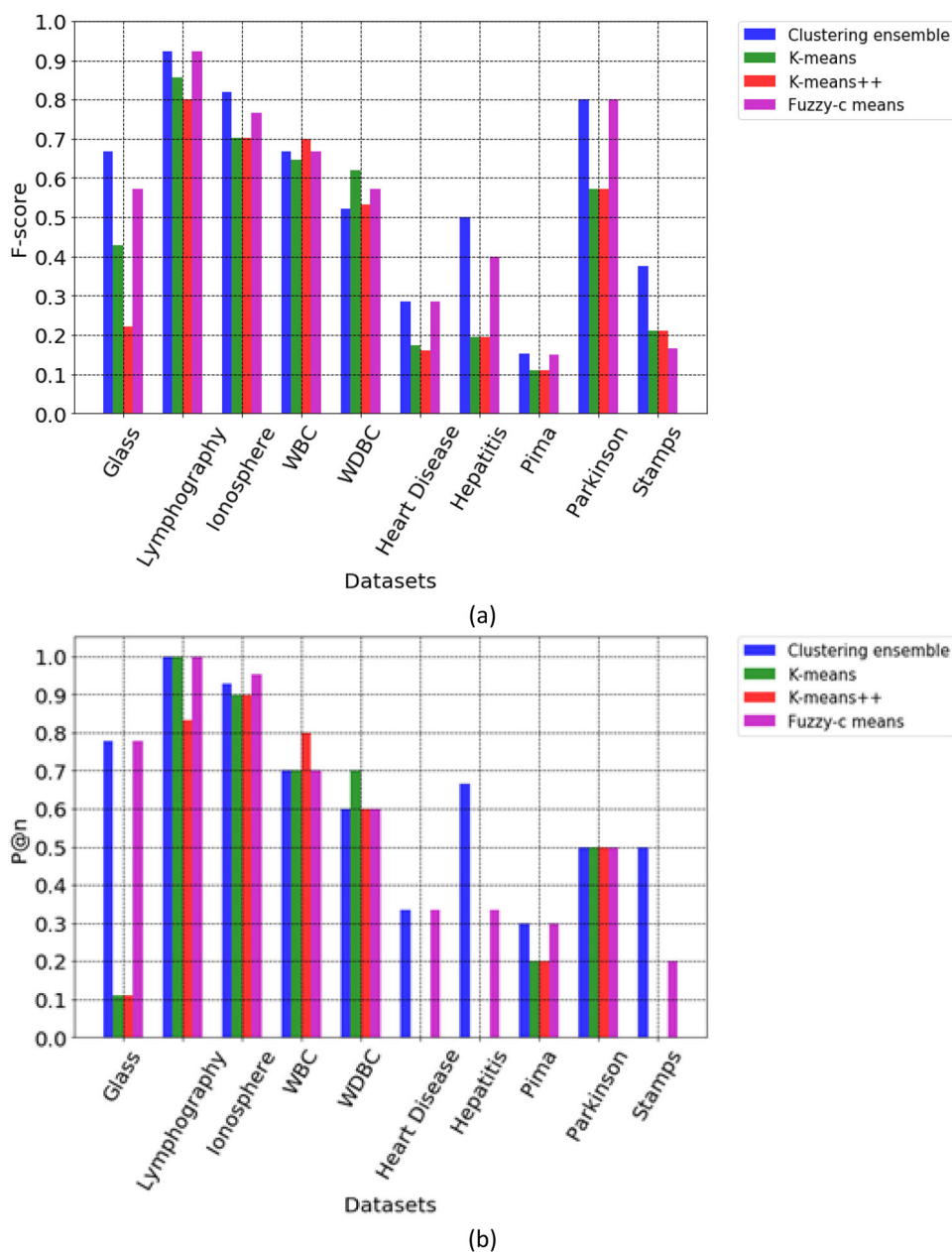


**Results on the database**

Results have been calculated for the mentioned 10 different datasets. We conduct multiple experiments by setting different parameters, i.e., by setting different cluster membership thresholds ( $P_{th}$ ) and varying distance threshold ( $d_{th}$ ) and the

number of clusters ( $K$ ) over a wide range of values. The parameter values for which we obtain the maximum F-score are considered as the best parameter values (see Table 2). Figure 3 shows the variation of the F-score with the cluster membership threshold ( $P_{th}$ ) parameter values for the present clustering ensemble method. In Table 2, the mentioned score

**Fig. 4** Performance of our proposed ensemble method in comparison with K-means, K-means++, and Fuzzy C-means clustering algorithms. The comparisons are made in terms of **a** F-score and **b**  $P@n$  for all datasets used here. In **b**, the missing bars indicate zero scores



is obtained using the best parameter values shown therein.  $P@n$  scores corresponding to the maximum F-score are considered in the results shown below. Using the parameter values (mentioned in Table 2), we compute the results in terms of F-score and  $P@n$  of the proposed algorithm and compare these results with those obtained using only K-means, K-means++, and Fuzzy C-means exclusively in place of the clustering ensemble. The comparative results are shown in Fig. 4. The results of this figure suggest that the present clustering ensemble method performs reasonably better as compared to K-means, K-means++, and Fuzzy C-means individually.

### Dunn Index variation for convergence of iteration

In this subsection, we show the Dunn index variation with each step of the iterative process until the convergence criterion is satisfied (i.e., till the Dunn index gets maximized, see section “Dunn index maximization for convergence of iteration”) for all of the algorithms (i.e., clustering ensemble, K-mean, K-means++, and Fuzzy C-means). The results for which are shown in Fig. 5.

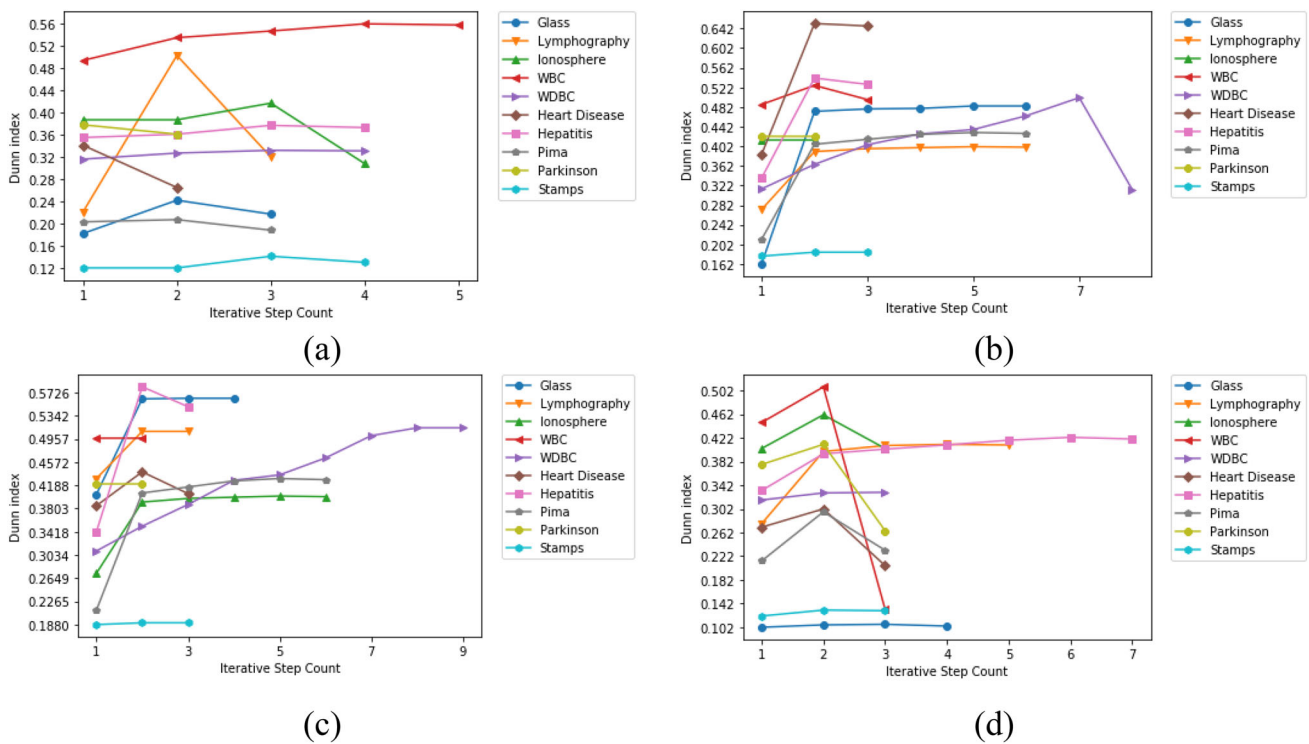


Fig. 5 Dunn index variation for all of the algorithms: **a** clustering ensemble, **b** K-means, **c** K-means++, and **d** Fuzzy C-means

Table 3 Performances in terms of P@n score of different base clustering algorithms and the alternative ensemble methods constructed with varying these base clustering algorithms

Method	Datasets									
	Glass	Lymphography	Ionosphere	WBC	WDBC	Heart Disease	Hepatitis	Pima	Parkinson	Stamps
<b>Base clustering models</b>										
KM	0.11	<b>1.00</b>	0.90	0.70	<b>0.70</b>	0.00	0.00	0.20	0.50	0.00
KM++	0.11	0.83	0.90	0.80	0.60	0.00	0.00	0.20	0.50	0.00
FCM	0.78	<b>1.00</b>	<b>0.95</b>	0.70	0.60	0.33	<b>0.33</b>	<b>0.30</b>	0.50	<b>0.20</b>
SOM	0.11	<b>1.00</b>	0.50	0.60	0.20	<b>0.67</b>	0.00	0.00	<b>1.00</b>	0.00
SL	0.11	<b>1.00</b>	0.58	<b>1.00</b>	0.67	0.00	0.00	0.30	0.50	0.00
<b>Iterative ensemble with distance-based data filtering-based clustering techniques</b>										
Method 1	0.44	<b>1.00</b>	0.50	0.50	<b>0.60</b>	<b>0.67</b>	0.33	0.10	0.50	0.00
Method 2	0.44	0.22	0.76	0.30	0.30	0.33	0.33	0.30	<b>1.00</b>	0.17
Method 3	0.44	0.21	0.74	0.10	0.20	0.33	0.33	0.29	<b>1.00</b>	<b>0.50</b>
Method 4	<b>0.80</b>	<b>1.00</b>	0.91	0.25	0.33	0.50	0.33	<b>0.50</b>	0.50	0.33
<b>Proposed method</b>	0.78	<b>1.00</b>	<b>0.93</b>	<b>0.70</b>	<b>0.60</b>	0.33	<b>0.67</b>	0.30	0.50	<b>0.50</b>

In this table, KM, KM++ , FCM, SOM, and SL stand for the algorithms: K-means, K-means++ , Fuzzy C-means, Self-Organizing Map, and Single-Linkage algorithms, respectively. Boldface numbers indicate the best scores (in each category)

### Performance of alternative ensemble-based outlier detection algorithms

In our proposed work, we have used three clustering algorithms: K-means, K-means++, and Fuzzy C-means. However, to test the performance of the proposed iterative ensemble with distance-based data filtering approach with

varying base clustering algorithms we have performed more experiments. For this, we have considered two more clustering techniques namely, Self-Organizing Map (SOM) [41], and Single-Linkage (SL) [42]. We have constructed 4 new iterative ensemble methods with distance-based data filtering approach with the help of the K-means, Fuzzy C-means, SOM, and SL clustering algorithms. The 4 different methods

**Table 4** Performances in terms of F-score of different base clustering algorithms and the alternative ensemble methods constructed with varying these base clustering algorithms

Method	Datasets									
	Glass	Lymphography	Ionosphere	WBC	WDBC	Heart disease	Hepatitis	Pima	Parkinson	Stamps
<b>Base clustering models</b>										
KM	0.42	0.86	0.71	0.65	0.62	0.17	0.19	0.12	0.57	<b>0.22</b>
KM++	0.22	0.80	0.71	0.70	0.53	0.16	0.19	0.12	0.57	<b>0.22</b>
FCM	<b>0.57</b>	0.92	<b>0.76</b>	0.67	0.57	<b>0.29</b>	<b>0.40</b>	<b>0.15</b>	<b>0.80</b>	0.17
SOM	0.14	<b>1.00</b>	0.53	0.73	0.24	0.12	0.17	0.11	0.67	<b>0.22</b>
SL	0.16	<b>1.00</b>	0.63	<b>0.75</b>	<b>0.63</b>	0.10	0.29	0.09	0.50	0.20
<b>Iterative ensemble with distance-based data filtering-based clustering techniques</b>										
Method 1	0.62	<b>1.00</b>	0.53	0.56	<b>0.60</b>	<b>0.57</b>	0.40	0.13	0.67	0.15
Method 2	0.62	0.62	0.84	0.35	0.14	0.40	0.50	0.30	0.80	0.40
Method 3	<b>0.67</b>	0.60	0.85	0.15	0.22	0.50	<b>0.67</b>	0.24	<b>1.00</b>	0.40
Method 4	0.62	0.62	<b>0.86</b>	0.14	0.25	0.40	0.57	<b>0.31</b>	0.80	<b>0.46</b>
<b>Proposed method</b>	<b>0.67</b>	0.92	0.82	<b>0.67</b>	0.52	0.29	0.50	0.15	0.80	0.38

In this table, KM, KM++, FCM, SOM, and SL stand for the algorithms: K-means, K-means++, Fuzzy C-means, Self-Organizing Map, and Single-Linkage, respectively. Boldface numbers indicate the best scores (in each category)

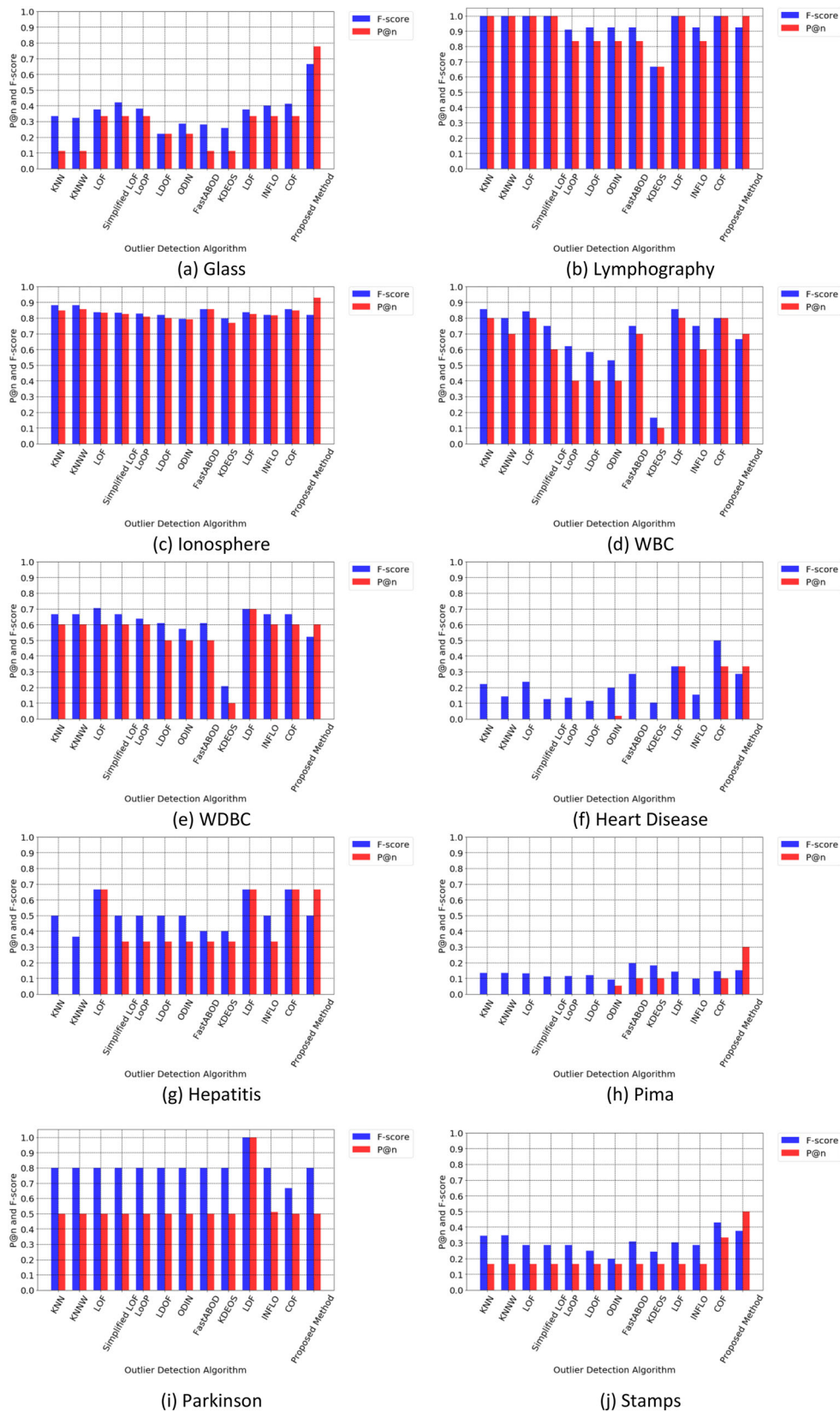
comprising 4 different clustering ensembles are Method 1 (K-means, SOM, and SL), Method 2 (K-means, SOM, and Fuzzy C-means), Method 3 (K-means, SL, and Fuzzy C-means), and Method 4 (SL, SOM, and Fuzzy C-means) have been built. The performances of the 4 alternative methods along with the proposed one (K-means, K-means++, and Fuzzy C-means) are shown in Table 3 (for  $P@n$ ) and Table 4 (for F-score). These tables also include the performances of K-means, K-means++, Fuzzy C-means, SOM, and SL. From the results it is clear that the performance of the proposed approach is better in most of the cases.

### Comparison with other outlier detection algorithms

We compare the performance of our proposed algorithm (iterative ensemble method with distance-based data filtering) with 12 other existing outlier detection algorithms (viz., kth-Nearest Neighbor (kNN), kNN-weight (kNNW), Local Outlier Factor (LOF), Simplified LOF, Local Outlier Probabilities (LoOP), Local Distance-based Outlier Factor (LDOF), Outlier Detection using Indegree Number (ODIN), Fast Angle-Based Outlier Detection (FastABOD), Kernel Density Estimation Outlier Score (KDEOS), Local-Density Factor (LDF), Influenced Outlierness (INFLO) and Connectivity-based Outlier Factor (COF)) the results. The score mentioned for the comparative methods is the experimental outcome of the work mentioned in [43]. The metrics  $P@n$  and F-score are only used for the comparative purpose as they are popularly used comparative metrics used for comparing outlier detection algorithms. These comparative results are shown in Fig. 6.

As indicative of the results shown in Fig. 6a, we observe that our proposed method outperforms the 12 other outlier detection algorithms on the Glass dataset and our proposed method also registers the highest  $P@n$  score and F-score. On the Lymphography dataset (see Fig. 6b for performance results) our proposed method performs comparably to the 12 other outlier detection algorithms. Our proposed method registers the highest  $P@n$  score along with a comparable F-score and gives better overall performance than LoOP, LDOF, ODIN, Fast ABOD, KDEOS, and INFLO algorithms. Following a similar trend to the previous results our proposed method registers the highest  $P@n$  score on the Ionosphere dataset (see Fig. 6c for performance results) and also shows comparable F-score performance. So, we infer that our proposed method outperforms all of the 12 outlier detection algorithms on the Ionosphere dataset. Only the KNN and KNNW algorithms (both are supervised outlier detection algorithms) show performance comparable to our proposed method on the Ionosphere dataset.

Figure 6d suggests that our proposed method gives better overall performance than the LoOP, LDOF, ODIN, and KDEOS algorithms on the WBC dataset but lags when compared to KNN and LDF algorithms both of which registered the highest F-score. Similar to what we observed before, our proposed method outperforms the KDEOS algorithm on the WDBC dataset and shows performance comparable to the LDF, ODIN, and FastABOD algorithms as evidenced by the results shown in Fig. 6e. The LOF algorithm registers the highest F-score on the WDBC dataset but is outperformed by the LDF algorithm on the  $P@n$  results. The performance results on the Heart Disease dataset are shown in Fig. 6f. We observe that the highest  $P@n$  scores are registered by our



**Fig. 6** Performance comparison of the proposed method with state-of-the-art techniques on the 10 datasets used in this work. The missing bars (in **f** and **h**) indicate zero scores

proposed method as well as by LDF and COF algorithms but our method falls behind in the F-score results registering the third highest score (behind COF and LDF algorithms) but in overall performance, our proposed method gives superior results to the rest of the 10 outlier detection algorithms. The results on the Hepatitis dataset shown in Fig. 6g again show that our proposed method shows comparable performance to the other algorithms. Our proposed method registers the highest  $P@n$  score along with the LDF, LOF, and COF algorithms.

Also, our proposed method registers the second highest F-score behind the 3 previously mentioned algorithms but shows greater overall performance than the rest of the 12 outlier detection algorithms on the Hepatitis dataset. On the Pima dataset (see Fig. 6h for performance results) our proposed method registers the highest  $P@n$  score and a comparable F-score. Only the FastABOD and KDEOS algorithms show similar performance to our method with FastABOD registering the highest F-score with KDEOS following in at a close second with our method falling in at third. So, we infer that our proposed method shows superior performance than the rest of the 10 outlier detection algorithms on the Pima dataset. Comparison results on the Parkinson dataset shown in Fig. 6i show that the LDF algorithm has the best  $P@n$  and F-score performance and all the other algorithms (including our proposed method) show exactly similar performance. Finally, in Fig. 6j, we show the performance results of the algorithms on the Stamps dataset. As we see from the comparison results that our proposed method again registers the highest  $P@n$  score with F-score comparable to what we observe for the other outlier detection algorithms. Only the COF algorithm shows performance comparable to our proposed method and registers the highest F-score. But all the other outlier detection algorithms fall behind in both the  $P@n$  score and F-score performance. Hence, our proposed method shows better overall performance than all the 12 other outlier detection algorithms on the Stamps dataset.

So, the comparison results indicate that our proposed algorithm shows good performance when compared to the said 12 existing outlier detection algorithms (both classification and clustering based) for all of the 10 machine learning datasets from our database. For some of the datasets, our algorithm gives outright better results than all the 12 algorithms whereas on other datasets it gives results comparable to those algorithms. We also infer that irrespective of the outlier type (isolated or clustered) present in the dataset our proposed algorithm has reasonable outlier detection accuracy as evidenced from the comparison results shown above.

## Conclusion

In this paper, we propose an algorithm that consists of an iterative clustering ensemble of K-means, K-means++, and Fuzzy C-means clustering algorithms and a distance-based filtering method for outlier detection from unlabeled datasets. We observe that using a clustering ensemble, we obtain results comparable or better than those obtained using individual clustering techniques (i.e., K-means, K-means++, and Fuzzy C-means) in place of the ensemble. Using an ensemble of clustering techniques, we can compensate for the erroneous results that we obtain from any of the individual clustering algorithms. The distance-based filter is effective in removing those outliers that remain undetected by our iterative clustering ensemble. Our results that we have obtained by experimenting on 10 datasets from various application domains infer that the proposed algorithm performs superior to some of the pre-established outlier detection algorithms (both classification and clustering based) on certain datasets. The performance of the proposed algorithm varies depending on the threshold values (cluster membership threshold  $P_{th}$  and distance threshold  $d_{th}$ ) chosen. In the future, an intelligent evaluation method can be designed to set the values of  $P_{th}$  and  $d_{th}$  for better estimation of the outlier data points. A larger clustering ensemble can be used which will combine results from multiple pre-established clustering outlier detection algorithms used for detecting outliers. In place of the distance-based data filtering scheme, a weighted method of data filtering can be used for the detection of possible outlier clusters which remain undetected by the clustering ensemble.

**Funding** The authors declare that they have not received any funds from any source to conduct this research.

**Availability of data and material** Not applicable in our case.

**Code availability** Not applicable in our case.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copy-

right holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Borah A, Nath B (2019) Rare pattern mining: challenges and future perspectives. *Complex Intell Syst* 5:1–23
- Dhieb N, Ghazzai H, Besbes H, Massoud Y (2019) A very deep transfer learning model for vehicle damage detection and localization. In: 2019 31st international conference on microelectronics (ICM). IEEE, pp 158–161
- Sarkar BK (2017) Big data for secure healthcare system: a conceptual design. *Complex Intell Syst* 3:133–151
- Shambharkar V, Sahare V (2016) Survey on outlier detection for support vector machine. *Int J Data Min Tech Appl* 5:11–14
- Shah V, Aggarwal AK, Chaubey N (2017) Performance improvement of intrusion detection with fusion of multiple sensors. *Complex Intell Syst* 3:33–39
- Carreño A, Inza I, Lozano JA (2020) Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. *Artif Intell Rev* 53:3575–3594
- Tian W, Liu J (2009) Intrusion detection quantitative analysis with support vector regression and particle swarm optimization algorithm. In: 2009 international conference on wireless networks and information systems. IEEE, pp 133–136
- Save P, Tiwarekar P, Jain KN, Mahyavanshi N (2017) A novel idea for credit card fraud detection using decision tree. *Int J Comput Appl* 161:6–9
- Aggarwal CC (2017) Proximity-based outlier detection. *Outlier analysis*. Springer, Berlin, pp 111–147
- Zhang J, Zulkernine M (2006) Anomaly based network intrusion detection with unsupervised outlier detection. In: IEEE international conference on communications
- Zhang K, Shi S, Gao H, Li J (2007) Unsupervised outlier detection in sensor networks using aggregation tree. In: International conference on advanced data mining and applications. Springer, pp 158–169
- Dasgupta D, Majumdar NS (2002) Anomaly detection in multidimensional data using negative selection algorithm. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). IEEE, pp 1039–1044
- Markou M, Singh S (2003) Novelty detection: a review—part 1: statistical approaches. *Signal Process* 83:2481–2497
- Saha A, Chatterjee A, Ghosh S et al (2021) An ensemble approach to outlier detection using some conventional clustering algorithms. *Multimed Tools Appl* 80:35145–35169. <https://doi.org/10.1007/s11042-020-09628-5>
- Hautamäki V, Cherednichenko S, Kärkkäinen I, et al (2005) Improving K-means by outlier removal. In: Scandinavian conference on image analysis. Springer, pp 978–987
- He Z, Xu X, Deng S (2003) Discovering cluster-based local outliers. *Pattern Recognit Lett* 24:1641–1650
- Hawkins S, He H, Williams G, Baxter R (2002) Outlier detection using replicator neural networks. In: Kambayashi Y, Winiwarter W, Arikawa M (eds) Data warehousing and knowledge discovery. DaWaK 2002. Lecture Notes in Computer Science, vol 2454. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-46145-0\\_17](https://doi.org/10.1007/3-540-46145-0_17)
- Loperfido N (2019) Kurtosis-based projection pursuit for outlier detection in financial time series. *Eur J Financ*. <https://doi.org/10.1080/1351847X.2019.1647864>
- Zhang K, Hutter M, Jin H (2009) A new local distance-based outlier detection approach for scattered real-world data. In: Theeramunkong T, Kijssirikul B, Cercone N, Ho TB (eds) Advances in knowledge discovery and data mining. PAKDD 2009. Lecture Notes in Computer Science, vol 5476. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-01307-2\\_84](https://doi.org/10.1007/978-3-642-01307-2_84)
- Bay S, Schwabacher M (2003) Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *Proc ACM SIGKDD Int Conf Knowl Discov Data Min*. <https://doi.org/10.1145/956750.956758>
- Ghoting A, Parthasarathy S, Otey ME Fast mining of distance-based outliers in high-dimensional datasets. In: Proceedings of the 2006 SIAM international conference on data mining. pp 609–613
- Jiang M-F, Tseng S, Su CM (2001) Two-phase clustering process for outliers detection. *Pattern Recognit Lett* 22:691–700. [https://doi.org/10.1016/S0167-8655\(00\)00131-8](https://doi.org/10.1016/S0167-8655(00)00131-8)
- Chen W, Tian Z, Zhang L (2020) Interpolation-based outlier detection for sparse, high dimensional data. *J Phys Conf Ser* 1437:12059. <https://doi.org/10.1088/1742-6596/1437/1/012059>
- Tsai C-F, Lin C-Y (2010) A triangle area based nearest neighbors approach to intrusion detection. *Pattern Recognit* 43:222–229. <https://doi.org/10.1016/j.patcog.2009.05.017>
- Wang C, Liu Z, Gao H, Fu Y (2019) Applying anomaly pattern score for outlier detection. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2895094>
- Feng Q, Zhang Z, Huang Z, Xu J, Wang J (2019) Improved algorithms for clustering with outliers. In: Proc. 30th International symposium on algorithms and computation (ISAAC 2019)
- Yang X, Zhang G, Lu J (2011) A kernel Fuzzy C-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises. *Fuzzy Syst IEEE Trans* 19:105–115. <https://doi.org/10.1109/TFUZZ.2010.2087382>
- Aggarwal C (2012) Outlier ensembles: position paper. *SIGKDD Explor* 14:49–58
- Kim E-Y, Kim S-Y, Ashlock D, Nam D (2009) MULTI-K: accurate classification of microarray subtypes using ensemble K-means clustering. *BMC Bioinform* 10:260. <https://doi.org/10.1186/1471-2105-10-260>
- Chen J et al (2017) Outlier detection with autoencoder ensembles. In: Proceedings of the 2017 SIAM international conference on data mining. Society for Industrial and Applied Mathematics
- Hartigan JA (1979) A K-means clustering algorithm: Algorithm AS 136. *Appl. Stat.* 28:126–130
- Lloyd S (1982) Least squares quantization in PCM's. *IEEE Trans Inf Theory* 28:129–136. <https://doi.org/10.1109/TIT.1982.1056489>
- Arthur D, Vassilvitskii S (2007) K-means++: the advantages of careful seeding. In: Proc. of the annu. ACM-SIAM Symp. on discrete algorithms. pp 1027–1035
- Bezdek J, Ehrlich R, Full W (1984) FCM—the Fuzzy C-means clustering-algorithm. *Comput Geosci* 10:191–203. [https://doi.org/10.1016/0098-3004\(84\)90020-7](https://doi.org/10.1016/0098-3004(84)90020-7)
- Dunn JC (2008) Well-separated clusters and optimal fuzzy partitions. *Cybern Syst* 4:95–104. <https://doi.org/10.1080/01969727408546059>
- Dunn J (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Cybern Syst* 3:32–57. <https://doi.org/10.1080/01969727308546046>
- Pal R, Yadav S, Karnal R (2020) EEWC: energy-efficient weighted clustering method based on genetic algorithm for HWSNs. *Complex Intell Syst* 6(2):391–400
- Malakar S, Sharma P, Singh PK et al (2017) A holistic approach for handwritten Hindi word recognition. *Int J Comput Vis Image Process* 7:59–78. <https://doi.org/10.4018/IJCVIP.2017010104>
- Järvelin K, Kekäläinen J (2017) IR evaluation methods for retrieving highly relevant documents. *ACM SIGIR Forum* 51:243–250. <https://doi.org/10.1145/3130348.3130374>

40. Manning C, Raghavan P, Schütze H (2010) Introduction to information retrieval. *Nat Lang Eng* 16(1):100–103
41. Kohonen T (1990) The self-organizing map. *Proc IEEE* 78:1464–1480
42. Seifoddini HK (1989) Single linkage versus average linkage clustering in machine cells formation applications. *Comput Ind Eng* 16:419–426
43. Campos G, Zimek A, Sander J et al (2016) On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Min Knowl Discov*. <https://doi.org/10.1007/s10618-015-0444-8>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.