Clustering-Based Outlier Detection Technique Using PSO-KNN

Sushilata D. Mayanglambam[1, 2*], Rajendra Pamula[1], and Shi-Jinn Horng[3]

[1] *Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad, Jharkhand-826004, INDIA*
[2] *Department of Computer Engineering, Mizoram University, Aizawl, Mizoram-796004, INDIA*
[3] *Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei-106335, TAIWAN*
[*] *Corresponding author. E-mail: sushi.2016dr1133@cse.iitism.ac.in;*

Received: Sept. 25, 2022; Accepted: Dec. 12, 2022

In this work, we present an unsupervised machine learning algorithm for outlier detection by integrating Particle Swarm Optimization (PSO) and the K-nearest neighbor (KNN) technique. Initially, the data clustering of the considered datasets was carried out using PSO to obtain optimized clusters. In the optimization process, we have adopted Davies-Bouldin (DB) index as a fitness function. The optimized clusters were pruned to exclude densely packed inliers data. Thereafter, the KNN method was employed to detect outliers present in the datasets. Our proposed algorithm was tested for outlier detection on eight different datasets and compared its performance with PSO+K-means, K-means, Local Outlier Factor (LOF), and Local Distance-based Outlier Factor (LDOF) methods. Our results show that the outlier detection efficiency of the proposed method outperforms than other four techniques. We believe that our proposed technique simple and efficient in finding the outliers in various types of datasets and it could be a promising tool for outlier detection in data mining.

Keywords: Particle Swarm Optimization; Davies-Bouldin Index; K-Nearest Neighbors; Outlier Detection

http://dx.doi.org/10.6180/jase.202312_26(12).0003

1. Introduction

The detection of abnormal data so-called outliers is of considerable importance and very essential in monitoring irregularities of information and decision-making of many real-world problems. Outlier detection is extensively used in a wide variety of domains such as banking, stock market, medical diagnosis, industry, computer networking, cyber security, etc [1, 2]. The nature of the outlier data pattern does not follow the well-defined definition of normal data. To accurately identify the abnormal patterns that deviate from the expected behaviors is a challenging problem for data mining researchers. In this regard, several outlier detection techniques have been proposed based on statistical, distance, density, clustering, learning, ensemble, graph, etc.[3, 4].

Outlier detection algorithms are generally based on either supervised learning or unsupervised learning. In the case of a supervised learning outlier detection algorithm, the datasets are neatly classified and labeled as normal and abnormal data. However, when a new anomaly with completely unknown characteristics [for example SARS-COV-2 (Covid-19) in human diseases] has appeared in the dataset, the supervised learning algorithm may fail because the algorithm identifies only the features that are explicitly labeled in the dataset. Whereas, an unsupervised outlier detection algorithm may be overcome because it can learn the inherent structure of the data without prior labeling of abnormalities in the dataset. But, in some cases, unsupervised learning may not be able to identify multiple outliers of different types that are present in the dataset [5]. Nevertheless, such a limitation can be overcome by improving the algorithm with new mathematical models. This suggests that the unsupervised learning technique for outlier detection could be a preferential choice over super-

vised learning when prior information of the outliers is not available to us. The clustering method has the advantage of grouping a set of data in unstructured datasets in such a manner that it maximizes the similarity of members in a cluster and minimizes the similarity of members between two different clusters [6]. There are several optimization algorithms including classical methods such as Broydon-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton algorithm, conjugate gradient methods, and new heuristic methods like genetic algorithm, differentia evolution, particle swarm optimization, any colony optimization, immune optimization algorithm, etc., [7, 8]. Particularly in data clustering problems, earlier reports have suggested that evolutionary, and swarm intelligence algorithms outperform the classical methods [9, 10].

Particle Swarm Optimization (PSO) is one of the swarm intelligence techniques, which has been extensively used in solving real-world optimization problems where traditional methods fail or have limitations [11]. Unlike evolutionary methods, PSO has few parameters to adjust and it does not involve complex operators like mutation, crossover, etc. found in genetic algorithm [12, 13]. In data clustering application, PSO does not require previous knowledge of the dataset to be clustered and it can find out the optimal number of clusters dynamically. Various PSO methods have been reported in the literature which adopted different fitness functions. However, there is no single algorithm including classical as well as heuristic methods that can cluster all the real-world datasets effectively and accurately without error [9]. Hence, a statistical-mathematical empirical function known as cluster validity index is defined, and based on this score, the clustering can be validated as good or poor. Particularly in data clustering using the PSO algorithm, a good validity index function has to be chosen to judge the fitness of the clustering. In recently reported work, DB was employed as a fitness function in the PSO algorithm, however, the study was limited to data clustering of a few chosen datasets [14].

In this work, we have explored the Davies-Bouldin (DB) index as a fitness function for validation of the clusters in the PSO algorithm, and the K-nearest neighbor (KNN) method was adopted to score the outliers in eight different benchmark datasets. The features that are only inherent to the dataset were considered for calculating the DB index [15, 16]. We have used DB index because its measures the average similarity between the clusters in the dataset by assigning a score, which was calculated from the ratio of within-cluster distances to between-cluster distances [17–19]. Furthermore, adopting DB as a fitness function in the PSO algorithm enables the generation of clusters with high intra-cluster similarity as well as low inter-cluster similarity [20]. KNN method could find the KNN of every data in the clusters and evaluate the outliers scores of each data in the cluster. Based on the outlier scores, we can identify whether the data has low-density neighbors or not, and finally, find out the outliers in the clusters. However, computing the KNN of each data in the clusters is tedious [21]. Hence, the data pruning method was implemented to exclude densely packed inliers data as well as reduced the complexity in the outlier detection. The main contribution of this work are:

- Adopted DB index as a fitness function in data clustering using PSO algorithm to obtain optimized centroids and determine better clusters.
- Implemented data pruning technique to remove densely populated inliers data points, which significantly reduces the time and space complexity.
- Simple KNN method was used to calculate the outlier score of the unpruned datapoints.
- Integrated PSO and KNN algorithm effectively determine outliers in most of the chosen datasets and its performance was superior than the other outlier detection methods.

2. Related work

Outlier detection is one of the extensive research domains in data mining. Particularly in outlier detection, various techniques have been reported in the earlier literature [1]. PSO was first introduced by Kennedy and Eberhart for optimizing non-linear function to simulate social behavior [11]. Over time, PSO has been significantly evolved and contributed to solving problems in many branches such as science, engineering, finance, marketing, biomedical, defense, networking, etc [22]. It is one of the most widely used optimization technique in data clustering [13, 23, 24]. In addition, PSO has been employed in anomaly/outlier detection problems. Merza et al. used the PSO algorithm to calculate the distance between data points to detect outliers [25]. They have compared PSO results with Local Outlier Factor (LOF) and it was found that PSO has superiority over the LOF method. Bamakan et al. have proposed PSO integrated with multiple criteria linear programming to enhance the accuracy of detection in attacks in the network system [26]. Further, Wang and Qin have proposed a hybrid method by combining PSO with a K-means clustering algorithm to improve the accuracy of anomaly detection. The hybrid anomaly detection method show improvement in outlier detection as compared to other algorithms [27].

Wahid and Rao has employed PSO to detect outliers in high-dimensional data [28]. They have calculated the distance between data points and closest neighbors to find the degree of the anomaly of data points. Then, applied PSO to the degree of anomaly according to a preset threshold value to find outliers. They have concluded that PSO results were more accurate and efficient as compared with High Contrast Subspaces for Density-based Outlier Ranking (HICS) algorithm. Guo et.al. has introduced the particle swarm optimization algorithm with quantum behavior (QPSO) to train the neural network in anomaly detection.It was found that neural network trained with the QPSO algorithm shows better performance over other algorithms such as the PSO algorithm and genetic algorithm [29]. Further, PSO has been combined with a k-means algorithm, self-organizing map, fuzzy c-means, etc. for outlier detection [30–33]. Alguliyev et al. [31] have proposed a weighted grouping method by integrating PSO and K-means algorithm for outlier detection. Their result shows that the combined algorithm improved the accuracy of detection outliers as compared to the K-means algorithm. Merza and AL-Anber [25] proposed a control chart technique using linear regression based on the particle swarm optimization (CCT-LR-PSO) algorithm for detecting outliers.The results show that CCT-LR-PSO was more accurate and superior in outlier detection over the normal distribution method. In the real scenario, distributions of data are mostly unknown so clustering-based outlier detection gives better performance than other methods. Here, we proposed an integrated PSO and KNN as a clustering-based algorithm for outlier detection.

3. Some preliminary concepts

In this proposed work, the PSO algorithm was adopted to carry out the clustering of the selected datasets and determine the optimized clusters using DB as fitness function. Other unsupervised clustering techniques such as PSO+K-means, K-means, LOF, and LDOF were also used to compare the efficiency of outlier detection. A brief discussion of PSO+K-means, K-means, LOF, and LDOF is described in the appendix section. The following section describes the theory of methods adopted in this work.

3.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a widely adopted optimization algorithm based on swarm intelligence, and multiple variants of the PSO algorithm were developed to handle various complex problems in science and engineering [34–37]. Particularly, PSO is extensively used in the area of data mining like data clustering, feature selection, forecasting, outlier/anomaly detection, etc due to its simplicity and high efficiency [23, 38–40]. Adopting the PSO algorithm enables us to find out the optimal centroids of clusters by minimizing the intra-cluster distance as well as maximizing the inter-cluster distance. Although several researchers have reported PSO algorithms for data clustering, the performance to find an optimal solution for a given problem depends on the fitness function employed in the algorithm. The advantage of PSO over other optimization methods is that users only have to optimize a few input parameters to perform the task [41]. First, it randomly initializes the particle's position and velocity. Then, the particles in PSO learned from past experiences similar to our brain neurons [42, 43]. Evaluate the fitness function and modify the velocities based on previous best and global best positions as per Eq. (1). Thereafter, update the particle's position using Eq. (2). Then, the process is iterated until maximum iterations or any other termination criteria. The variables used in Eqs. (1) and (2) are defined in Table SA of the supplementary information.

$$V_{i+1} = \omega V_i + C_1 * r() * (Pb_i - X_i) \tag{1}$$

$$X_{i+1} = X_i + V_{i+1} \tag{2}$$

Further, we have implemented a restriction on velocity and position to move the lazy particles as well as to limit energetic particles movement within the search space [44]. This restriction helps faster convergence and better solution of the PSO algorithm.

3.2. K-nearest Neighbor

K-nearest neighbor is a simple machine learning non-parametric algorithm based on the supervised learning technique [45]. It has been used in applications like pattern recognition, data mining and intrusion detection, etc. In this algorithm, every unpruned data element contributed from all the clusters were considered, and calculated its euclidean distance. Thereafter, the euclidean distance of all the unpruned data was sorted in ascending order. Subsequently, the average K-nearest neighbor (KNN) distance of each data point to all the other data points was calculated using the following equation [46]:

$$d_r = \frac{1}{k} \sum_{s \in N_r} dist(r, s) \tag{3}$$

where N_r be the set of KNN of object r (excluding r), dist(r, s) is the distance between objects r and s. The KNN distance of r is given by the average distance of all objects from r in N_r.

4. Proposed integrated pso and knn algorithm

In this proposed work, we have integrated PSO and KNN methods for outlier detection. Here, the PSO algorithm facilitates optimizing the dataset to obtain the best clusters. We have used the DB index as an objective function for the validation of clusters. The DB index can be computed using the following [47]:

- Measured of scatteredness within the cluster C_i:

$$P_i = (\frac{1}{T_i} \sum_{n=1}^{T_i} |D_n - c_i|^t)^{\frac{1}{t}} \tag{4}$$

- Separation distance between two clusters C_i and C_j:

$$Q_{i,j} = (\sum_{k=1}^{m} |D_{n,i} - D_{n,j}|^q)^{\frac{1}{q}} \tag{5}$$

- Goodness of the clustering scheme of clusters C_i and C_j:

$$R_{i,j} = \frac{P_i + P_j}{Q_{i,j}} \tag{6}$$

- DB Index:

$$DB = \frac{1}{m} \sum_{i=1}^{m} D_i, where\ D_i \equiv \max_{j \neq i} R_{i,j} \tag{7}$$

where T_i is the number of data elements in the cluster C_i with centroid c_i, D_n is an n-dimensional feature vector assigned to cluster C_i, t represents the distance metrics. When $q = 2$, $Q_{i,j}$ gives the Euclidean distance function between the individual feature vectors $D_{n,i}$, and $D_{n,j}$ of clusters C_i and C_j respectively, and m is the number of clusters. The optimized clusters generated by the PSO are subsequently considered for data pruning. In the pruning process, the data points that are densely packed within every cluster were excluded as inliers. Here, we have calculated the distance of each data points to its centroid in a cluster, and if their distances are less than the average distance of all data points in the cluster (defined as the radius of the cluster), then those data points are pruned. This significantly reduces the complexity of the problem as well as reduces the computational cost and time. To identify the outliers among unpruned data points, we have employed the KNN algorithm. KNN algorithm is a simple, fast, and effective tool for finding outliers because the algorithm finds the neighbors of each data point based on Euclidean distance. Thereafter, each data point are sorted in the descending order based on their respective distance value. Then, the first top n data points with the highest Euclidean distance value are considered as outliers of the chosen dataset. The pseudo-codes of the proposed PSO-KNN algorithm used in the experimental analysis for outlier detection are described below. The pseudo-code of data clustering is given in Algorithm 1 and outlier detection is given in Algorithm 2.

Algorithm 1. Clustering algorithm

Data: Data, number of clusters, PSO parameters
Result: Optimized Centroids, **gBest**
for *each particle* **do**
  Initialize particle's position and velocity randomly
  Calculate fitness value $F(i,t)$ corresponded to position $X(i,t)$ using DB index as a fitness function $\%i,t$: particle i at iteration t.
  Assign **pBest and gBest**
end
while *maximum iteration or convergence criteria is not met* **do**
  for *each particle* **do**
    Calculate particle velocity using Eq. (1) and apply velocity limit
    Update particle position using Eq. (2) and apply position limit
  end
  Calculate fitness value F(i,t) corresponded to position X(i,t) using DB index as a fitness function
  Update **pBest and gBest**
end
Return optimized centroids, **gBest**

Algorithm 2. Outlier detection algorithm

Data: Data **X**
Result: Top-o Outliers in **X**
Using algorithm 1 to obtain optimized centroids
for *each x belong to X* **do**
  Group the data into clusters using the optimized centroids
  Compute radius of each cluster centroid
  Prune those data points for a distance less than its radius
  Compute KNN of unpruned points using the equation:
  $d_r = \frac{1}{k} \sum_{s \in N_r} dist(r,s)$
end
Sort KNN (unpruned points) in descending order.
Output Top-o Outliers.

5. Experiment details

We have considered eight different types of datasets for outlier detection using our proposed algorithm. Seven

datasets namely Forest fires, Ionosphere, Wisconsin Breast Diagnostic Cancer (WDBC), Yeast, E. coli, Letter, and Cardio were obtained from public repositories [48–51], and one synthetic data [52]. In our proposed PSO, the parameters which were taken in the algorithm were the maximum number of iterations = 400, number of particles = 50, personal learning coefficient = 1.5, global learning coefficient = 2, inertia weight = 1, inertia weight damping ratio = 0.99 and number of clusters = 3. The clustering was also performed for different values of PSO parameters including personal learning coefficient, global learning coefficient, inertia weight, and inertia weight damping ratio in one of the datasets, Forest Fires. It was found that the PSO parameter values mentioned above have the lowest DB index value, which is listed in Table 1. The efficiency of our proposed algorithm is compared with other outlier algorithms such as PSO+K-means, K-means, Local Outlier Factor (LOF), and Local Distance-Based Outlier Factor (LDOF) [20, 31, 53–56]. To evaluate the performance of the algorithms, Precision, Recall, Average-Precision and F1-Score were calculated. Thereafter, the Precision-Recall curve (PR curve) was plotted for all five methods. Here, if there are t true outliers in a data set and if an algorithm can detect n_t outliers among top n points then these are termed as true positive (TP). The true outliers that cannot be detected by an algorithm are false negative (FN), and t is defined as t = TP + FN. If the algorithm reports some inliers in top n points then these are false positive (FP), and n is defined as n = TP + FP. The precision and recall were calculated using the following equations [57]:

$$Precision = \frac{TP}{TP + FP} = \frac{n_t}{n} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} = \frac{n_t}{t} \tag{9}$$

The forest fire dataset is continuous data consisting of 517 data elements and each data element has 13 attributes[58]. The 13^{th} attribute corresponds to forest area under fire and content both least and significantly affected area. In our experiment, we have considered the most significantly burned areas data points as outliers and the rest of other data points as inliers, out of which 10 data elements were randomly selected as outliers. Johns Hopkins University Ionosphere dataset consists of 351 instances and it has a total of 35 attributes [49]. Out of which 34 attributes were continuous data and 35^{th} attribute described the status of signal reflected from the free-electron targets in the ionosphere, which are labeled as good (if signals are reflected) or bad (if signals are not reflected). We have considered the "bad" radar responses as outliers and 10 numbers of such instances were randomly pickup to be treated as outliers in the dataset. Wisconsin Breast Diagnostic Cancer (WDBC) dataset consists of 569 medical diagnosis records, each with 32 attributes(ID, diagnosis, 30 real-valued input features) [59]. The diagnosis data is binary, which is categorized as either benign or malignant. We treated the diagnosis data with labeled "benign" as normal data and those labeled "malignant" as outliers. In our experiment, we have used 357 Benign diagnosis records as normal data and randomly selected 10 malignant diagnosis data added into normal data as outliers. In the yeast dataset, there are 1484 instances and 8 attributes [51, 60]. Out of this, we have taken 1433 datapoints as normal data after removing duplicate data and randomly chose 10 datapoints of peroxisomal (POX) class were added as outliers. The resulting dataset of size 1443 was used in the experiment. E.coli dataset consists of 336 data points, and 7 attributes [61]. Out of 8 classes, 3 classes i.e. omL (outer membrane lipoprotein), imL (inner membrane lipoprotein), and imS (inner membrane, cleavable signal sequence) consisting of 9 data points were treated as outliers. However, in our experiment, we have randomly chosen only 5 outliers from 9 data points and added 327 normal data points. hence, the total size of the dataset used in our experiment was 332. The multi-dimensional text recognition dataset called the letter dataset has 1500 normal data points and 100 outliers [50]. After deleting duplicate data from the dataset, 55 outliers were randomly chosen from a pool of 100 outliers and added to the 1498 normal data points. As a result, we used 1553 data points altogether for this experiment. The Cardiotocography (Cardio) dataset is the measure of fetal heart rate and uterine contraction features recorded on cardiotocograms [62]. The dataset was classified as normal, suspect, and pathologic. The pathologic class consider as outliers and the suspected class was not considered in the prepared dataset. It consists of a total of 1831 data points, out of which 1655 are normal data and 176 are outliers datapoints. In our experiment, 1647 data points were considered as normal data after removing duplicate data points, 50 outliers were randomly selected and added into the normal data, which gave the total size of our dataset to be 1697. The synthetic dataset was generated using a Matlab function as described in the earlier reported literature [52]. The dataset consist of 200 data elements and each data element has 2 features. The whole dataset is grouped into 5 clusters, out of which three clusters are the majority and two are minority clusters. In the experiment, the three majority clusters that consist of a total of 173 data elements are considered as inliers. Ten data elements out of the remaining 27 data elements that belong to two minority clusters are randomly selected as outliers. All computation were performed using custom

Table 1. Comparison of DB index value for different PSO parameters in Forest Fires Dataset.

Inertia Weight	Inertia weight damping ratio	Personal learning coefficient	Global learning coefficient	Number of clusters	DB Index value
1	**0.99**	**1.5**	**2**	**3**	**0.5363**
0.5	0.5	1	1	3	0.7910
0	0	4	4	3	0.5541
1	1	2	2	3	0.5472

MATLAB program (MATLAB version R2014a). The program was run using HP workstation Z4 equipped with Intel Zenon processor with 32GB RAM. Table 2 summarizes the details of real and synthetic datasets like number of instances, features, and number of outliers used for our experiment analysis.

Table 2. Description of real and synthetic datasets used for experiment analysis.

Datasets	Instances including outliers	Attributes	Number of outliers
Forest Fire	511	11	10
Ionosphere	235	34	10
WDBC	367	30	10
Yeast	1443	8	10
E.Coli	332	7	5
Letter	1553	32	55
Cardio	1697	21	50
Synthetic	183	2	10

6. Results and discussion

At first, we have computed precision (Pr), recall (Re), average precision and F1-score for five outlier detection methods. High precision and high recall are very desirable to determine the optimal solution for finding the outliers present in a given dataset. This was estimated by setting different threshold values (which in our case is the different values of top n possible outliers) and determining their corresponding precision and recall. Thereafter, we have calculated average precision and plotted the corresponding precision-recall curve for all five methods. In our experiment, KNN (k) values were taken proportionately within 5% of the dataset size.

In the forest fire dataset, we randomly picked 10 outliers and added them to 501 normal data. The resulting dataset of size 511 was used in the experiment. The KNN (k) values of 5, 10 and 25 were used for calculating precision and recall. The results obtained at different threshold values of outliers i.e., n= 5, 10, 15, 20, 25, and 40 for k=10 are provided in Table 3, and for k=5, and 25 are provided in Table S1(a) of supplementary information. It can be seen from Table 3 that when the threshold value was set to top five possible outliers, our proposed PSO enables to detect 5 out of 10; LOF can detect 4 outliers out of 10; LDOF can detect 2 outliers out of 10; PSO+K-means and K-means algorithms can detect only 1 outlier out of 10. As we set the threshold value to top 25 possible outliers, our proposed algorithm detect all 10 outliers, whereas LOF detect 9 outliers, PSO+K-means and K-means can detect 8 outliers each, LDOF only detect 5 outliers out of total 10 outliers present in the dataset. When we set lower KNN value i.e., k=5, our proposed method could detect all 10 outliers, while other methods were not able to detect all 10 outliers even at the maximum threshold value. On the other hand, at larger KNN value i.e. K=25, the performance of LOF and LDOF were improved and detect all the 10 outliers at the threshold value beyond 15. Whereas our proposed method could detect 10 outliers at the maximum threshold value of 40. However, choosing a larger value of KNN value also opens the possibility of selecting outliers which are close to the inlier data points. Nevertheless, the overall performance of our proposed method at all KNN values was much higher than the other methods in this type of dataset. Further, the F1-score and average precision calculated for all five methods for k=10 are given in Tables 4 and 5 respectively. For F1-score of other k-values of 5 and 25 are provided in Table S1(a) of the supplementary information. It can be seen that our proposed PSO has the highest F1-score of 0.70 for n=10, and also the average precision of 0.5444 was found, which is highest among all five methods. The precision-recall (PR) curves of all five methods for k=10 is shown in Fig. 1, and for k=5 and 25 are provided in figure S1 of the supplementary information. It is clear from Fig. 1, and figure S1 that the PR curves for our proposed PSO has the largest area under precision-recall (PR) curve among all methods. And among all k-values, k=10 has the highest area under PR curve in our proposed method. Hence, a k value of 10 will provide the optimal solution to detect the outliers in such type of dataset.

In the Ionosphere dataset, we randomly picked 10 outliers and added them to 225 normal data. The resulting dataset of size 235 was used in the experiment. Precision and recall values for k= 3, 7, and 10 were calculated for all

Table 3. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF and Proposed PSO for k = 10 of Forest fires dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
5	1	0.2	0.1	1	0.2	0.1	4	0.8	0.4	2	0.4	0.2	5	1	0.5
10	4	0.4	0.4	4	0.4	0.4	6	0.6	0.6	2	0.2	0.2	7	0.7	0.7
15	6	0.4	0.6	6	0.4	0.6	7	0.47	0.7	2	0.13	0.2	7	0.47	0.7
20	6	0.3	0.6	6	0.3	0.6	7	0.35	0.7	3	0.15	0.3	9	0.45	0.9
25	8	0.32	0.8	8	0.32	0.8	9	0.36	0.9	5	0.2	0.5	10	0.4	1
40	9	0.23	0.9	9	0.23	0.9	10	0.25	1	5	0.125	0.5	10	0.25	1

Table 4. F1-score of Proposed PSO, PSO+K-means, K-means, LOF,and LDOF for k=10 in Forest fire dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
5	0.133	0.133	0.533	0.266	0.666
10	0.40	0.40	0.60	0.20	0.70
15	0.48	0.48	0.56	0.16	0.56
20	0.40	0.40	0.466	0.20	0.60
25	0.457	0.457	0.514	0.285	0.571
40	0.36	0.36	0.40	0.20	0.40

Table 5. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 5, 10 and 25 of Forest fire dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=5	0.5078	0.2497	0.2579	0.3622	0.0619
k=10	0.5444	0.3075	0.3075	0.4711	0.2014
k=25	0.5244	0.3744	0.3370	0.5694	0.5694

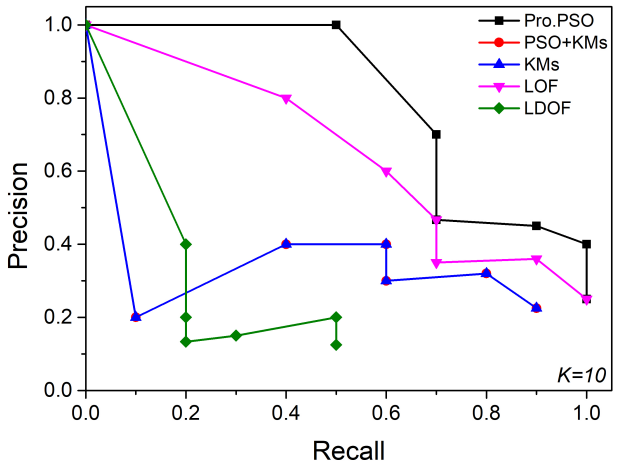

Fig. 1. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=10 for Forest fire dataset.

five methods. Table 6 listed the precision and recall values of k=7. In the experiment, different threshold values i.e., n= 5; 10; 15; 25; 40 and 60 were set in the calculation. It can be seen from Table 6 that our proposed PSO could find out all 10 outliers when the threshold value of possible outliers was at 40, while other two methods could detect only 8 to 9 outliers. The calculated precision and recall values for k=3 and k=10 are provided in the Table S2(a) of the supplementary information. The average precision of all the methods for different k-values are summarized in Table 8. It was found that the average precision of our proposed method was highest among all five methods, and k=10 yield the highest average precision value of 0.5047. The calculated F1-score of k=7 for all the methods are provided in Table 7, and similarly F1-scores for k=3 and 10 are provided in Table S2(b) of the supplementary information. We observed that our proposed PSO has the highest F1-score for all k-values among all the methods. In addition, the PR curve plotted for k=7 is shown in Fig. 2, and similar PRC for k=3 and k=10 are provided in figure S2 of the supplementary information. It is clear from Fig. 2 that the PR curves for our proposed PSO has larger area under curve than the other four methods at all k-values. Therefore, we expected that k-value of 10 will be suited to detect the outliers for this type of dataset.

In Wisconsin Breast Diagnostic Cancer (WDBC) dataset, it has 367 data that include 10 outliers and 357 normal data. The precision and recall were calculated at different threshold values n= 5, 10, 15, 20, 25, and 30 for k-values 5, 10, and 15 respectively. Table 9 shows the comparison

Table 6. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF, and proposed PSO for k = 7 of Ionosphere dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
5	5	1	0.5	5	1	0.5	4	0.8	0.4	4	0.8	0.4	5	1	0.5
10	6	0.6	0.6	7	0.7	0.7	7	0.7	0.7	5	0.5	0.5	7	0.7	0.7
15	8	0.53	0.8	8	0.53	0.8	7	0.47	0.7	6	0.4	0.6	8	0.53	0.8
25	8	0.32	0.8	8	0.32	0.8	8	0.32	0.8	8	0.32	0.8	8	0.32	0.8
40	9	0.23	0.9	9	0.23	0.9	8	0.2	0.8	8	0.2	0.8	10	0.25	1
60	9	0.15	0.9	9	0.15	0.9	8	0.13	0.8	9	0.15	0.9	10	0.17	1

Table 7. F1-score of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k=7 in Ionosphere dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
5	0.666	0.666	0.533	0.533	0.666
10	0.60	0.70	0.70	0.50	0.70
15	0.64	0.64	0.56	0.48	0.64
25	0.457	0.457	0.457	0.457	0.457
40	0.36	0.36	0.32	0.32	0.40
60	0.257	0.257	0.228	0.257	0.285

Table 8. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 3, 7 and 10 of Ionosphere dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=3	0.5033	0.495	0.4922	0.3883	0.2889
k=7	0.4950	0.4880	0.4714	0.4367	0.395
k=10	0.5047	0.4839	0.4811	0.4006	0.3839

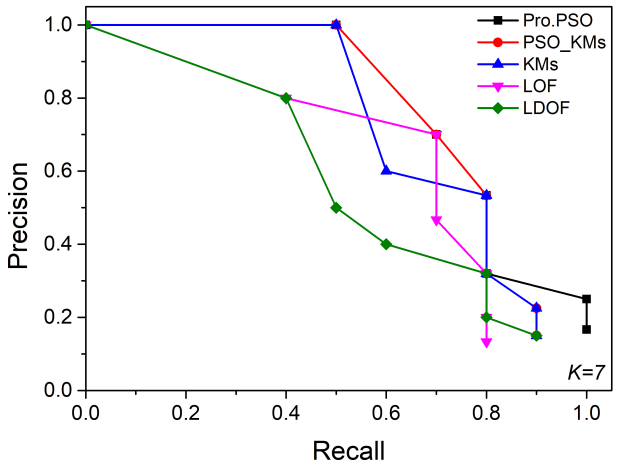

Fig. 2. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=7 for Ionosphere dataset.

of precision and recall for K-means, PSO+K-means, LOF, LDOF and proposed PSO for k = 15. As seen from Table 9, all 10 outliers were able to detect with our proposed PSO at threshold values of 10, whereas other methods could detect up 8 to 9 outliers. When the threshold set value increases to 15, the proposed PSO, PSO+K-means, LOF and K-means methods could find out all the 10 outliers present in the dataset, whereas LDOF can detect only 3 outliers. The calculated results of precision and recall for k=5 and k=10 are provided in Table S3(a) of the supplementary information. Table 11 provides the calculated average precision values for all three methods for k=5, 10, and 15 respectively. Interestingly, it can be seen that when KNN values were set at a larger value (for example k=15), the average precision of our proposed method was 0.650, which outperforms over other four methods. Both PSO+K-means and K-means methods have average precision of 0.6167 for k=15. Also, the average precision was found to be gradually increased with increasing the KNN values in all the methods. Further, the F1-scores for all the methods for k=15 are tabulated in Table 10, and the results of other k-values are provided in Table S3(b) of the supplementary information. A higher F1-score of our proposed was observed in all the k-values, indicating its consistency and superior performance as compared to other methods. Fig. 5 shows the PR curves plotted for k=15 for all five methods, and PR curves for other k-values of 3 and 10 are shown in figure S3 of the supplementary information. As we seen from the Fig. 3 that our proposed method has the largest area under the curve in comparison to the other methods for

all the k-values. For k=15, our proposed technique could achieved 100% precision and 100% recall, indicating that the best method for finding the outliers as compared to other techniques taken into consideration.

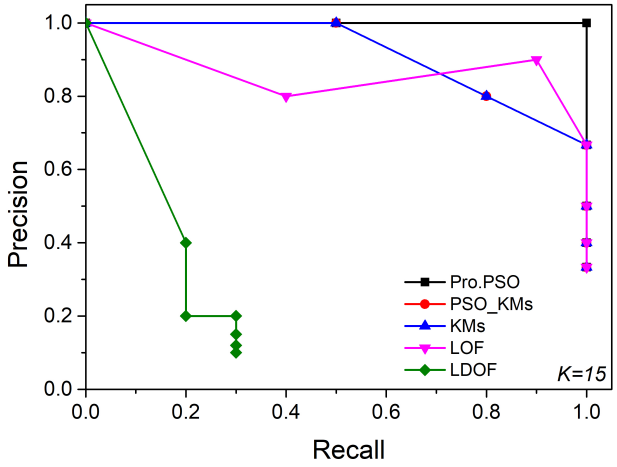

Fig. 3. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=15 for WDBC dataset.

In the synthetic dataset, 10 outliers randomly picked up from two small clusters were added to three larger clusters consisting of 173 normal data. The resulting dataset of size 183 was used in the experiment. The precision and recall were calculated for all the methods at different threshold values of k=5 are listed in Table 12. Similarly, we have calculated the precision and recall values for k=3 and k=10, which are provided in Table S4(a) of the supplementary information. As we see from the Table 12 that our proposed method can find out all 10 outliers at threshold value n $\geq$ 15, whereas PSO+K-means can detect 6 outliers, K-means can identify 9 outliers, LOF can find out 5 outliers respectively. Among all methods, LDOF can not identify any of the outlier. When the threshold value was set up to 60 possible outliers, all methods except LDOF can detect all 10 outliers, and LDOF can detect only 2 outliers. At k=10, our proposed method can detect all the outliers at lower threshold value. Table 14 summarized the calculated average precision of all methods discussed in this work. As we can see from Table 14 that the average precision values obtained for our proposed method was the highest at k=3, 5 and comparable to LOF at k=10. The highest average precision values of 0.5306 and 0.5639 were found for k=5, and k=10 respectively by our proposed method. Also, the average precision improves with increasing KNN values for all the methods. The calculated F1-scores for all the methods were tabulated in Table 13 for k=5, Table S4(b) for k=3, and 10 respectively. For k=5, the F1-score is highest for our proposed method, and comparable to other methods at k=3 and 10. Fig. 4 illustrates the PR curves of all the methods for k=5, and it is clearly seen that our proposed method has the highest area under the PR curve. Similar trend was found for other k-values of 3 and 10, which were shown in figure S4 of the supplementary information. These results reflect the superior performance of our proposed method over other methods. It has to be noted that although all k-values enables to detect outliers at different threshold values, a low threshold value of possible outliers will be beneficial to eliminate possible inliers predicted from our experiments. In addition, a lower k-value will reduce the time and complexity of the problem. Hence, we expected the KNN value of 5 at lower threshold value will give the best performance to detect outliers such type of synthetic dataset.

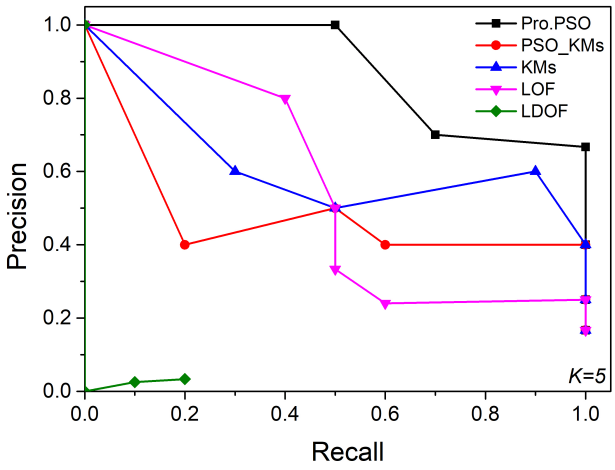

Fig. 4. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=5 for Synthetic dataset.

In the yeast dataset, there are 10 outliers and 1433 normal data points. The resulting dataset of size 1443 was considered in the experiment. The precision and recall were calculated for different k-values of all the algorithms at different threshold values. The result of k=10 is provided in Table 15, while the results of k=5 and 30 are provided in Table S5(a) of the supplementary information. It was found the performance of LOF and LDOF were very low for identifying the outliers for KNN value of 5. But, the results were improved at k=10. The performance of the K-means, PSO+K-means and our proposed methods were equal at lower KNN value (k=5). The performance of our method get improves over K-means and PSO+K-means when we increased the KNN value up to 10. Further, when we consider a higher KNN value i.e. k=30, interestingly the performance of LOF was as par with our proposed method.

Table 9. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF, and Proposed PSO for k = 15 of WDBC dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
5	5	1	0.5	5	1	0.5	4	0.8	0.4	2	0.4	0.2	5	1	0.5
10	8	0.8	0.8	8	0.8	0.7	9	0.9	0.9	2	0.2	0.2	10	1	1
15	10	0.67	1	10	0.67	0.8	10	0.67	1	3	0.2	0.3	10	0.67	1
20	10	0.5	1	10	0.5	0.8	10	0.5	1	3	0.15	0.3	10	0.5	1
25	10	0.4	1	10	0.4	1	10	0.4	1	3	0.12	0.3	10	0.4	1
30	10	0.33	1	10	0.33	1	10	0.33	1	3	0.1	0.3	10	0.33	1

Table 10. F1-score of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k=15 in WDBC dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
5	0.666	0.666	0.533	0.266	0.666
10	0.80	0.80	0.90	0.20	1.0
15	0.80	0.80	0.80	0.24	0.80
20	0.666	0.666	0.666	0.20	0.666
25	0.571	0.571	0.571	0.171	0.571
30	0.50	0.50	0.50	0.15	0.50

Table 11. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 5, 10 and 15 of WDBC dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=5	0.5889	0.5333	0.5333	0.0689	0.0689
k=10	0.6333	0.5833	0.5833	0.3033	0.1270
k=15	0.650	0.6167	0.6167	0.60	0.195

Table 12. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF, and proposed PSO for k = 5 of the Synthetic dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
5	3	0.6	0.3	2	0.4	0.2	4	0.8	0.4	0	0	0	5	1	0.5
10	5	0.5	0.5	5	0.5	0.5	5	0.5	0.5	0	0	0	7	0.7	0.7
15	9	0.6	0.9	6	0.4	0.6	5	0.33	0.5	0	0	0	10	0.67	1
25	10	0.4	1	10	0.4	1	6	0.24	0.6	0	0	0	10	0.4	1
40	10	0.25	1	10	0.25	1	10	0.25	1	1	0.025	0.1	10	0.25	1
60	10	0.17	1	10	0.17	1	10	0.17	1	2	0.03	0.2	10	0.17	1

Table 13. F1-score of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k=5 in Synthetic dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
5	0.40	0.266	0.533	0	0.666
10	0.50	0.50	0.50	0	0.70
15	0.72	0.48	0.40	0	0.80
25	0.571	0.571	0.342	0	0.571
40	0.40	0.40	0.40	0.04	0.40
60	0.285	0.285	0.285	0.057	0.285

However all the methods could detect only 6 out of 10 outliers at maximum k-value of 30 and maximum threshold value of 40. The calculated F1-score of k=10 for all the methods are summarized in Table 16 and the other k-values of 5 and 30 are provided in Table S5(b) of the supplementary information. Our proposed method achieved a maximum F1-score of 0.533 when k=30 was considered and threshold was set at 5. At higher threshold values, F1-scores were reduced to 0.24, and similar trend was found for other methods too. Table 17 provides the average precision of all methods at all k-values. As we see from the Table that our proposed method has the highest average precision at

Table 14. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 3, 5 and 10 of Synthetic dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=3	0.3161	0.2361	0.2728	0.0247	0.2211
k=5	0.5306	0.3528	0.4194	0.3817	0.0097
k=10	0.5639	0.4861	0.4861	0.5806	0.0964

k-values of 5 and 10. At k=30, our proposed method and LOF has almost equal average precision. The PR curves for all methods are shown in Fig. 5 for k=10, and for other k-values of 5 and 30 are provided in figure S5 of the supplementary information. Also, our proposed method has the highest area under PR curves among all methods.These results indicate that the overall performance of our proposed PSO method is better as compared to other four methods adopted in this work.

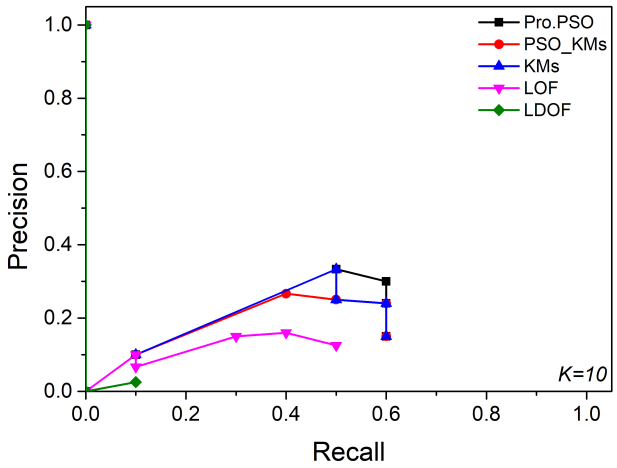

Fig. 5. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=10 for Yeast dataset.

Further, we consider E.coli dataset consisting of 5 distinct outliers and 327 inliers data points in our experiment. The precision and recall were calculated for all five methods at different threshold values for k=5, which are provided in Table 18 and for other k-values, they are provided in figure S6 of the supplementary information. It was found that only 4 outliers can be detected with all methods even after the threshold value was set up to 60 for all the k-values. At low threshold value i.e. k=5, our proposed method and PSO+K-means has the similar performance. However, at higher threshold value of k=20, all methods except LDOF exhibits almost equal performance. LDOF has very poor output in such type of dataset. Table 19 listed the F1-score of all methods for k=5, and F1-score of other k-values are provided in Table S6(b) of the supplementary information. The F1-score of our proposed method is slightly lower as compared to K-means method at lower k-values of k=3 and k=5, however, its score is comparable to K-means methods at higher k-value of 20. Further, the overall average precision calculated for all the methods are summarized in Table 20. It can be clearly seen that K-means method has the highest average precision value among all the methods at lower k-values and its value equals to PSO+K-means at k=20 in such type of dataset. This is expected because a large distinction of outliers from the inliers data can be easily predicted by the classical K-means technique as compared to PSO. We believed that the data pruning strategy adopted in our proposed algorithm might be inefficient when the size of the dataset is small and outliers are very distinct from the inliers. Nevertheless, our proposed PSO method can detect the equal number of outliers detected by K-means and other methods. Fig. 6 provides PR curves of all the methods for k=5 and similar PR curves for k=3 and k=20 are provided in figure S6 of the supplementary information. For the figures, it was found area under PR curve was highest for the k-means method in such dataset among all the methods.

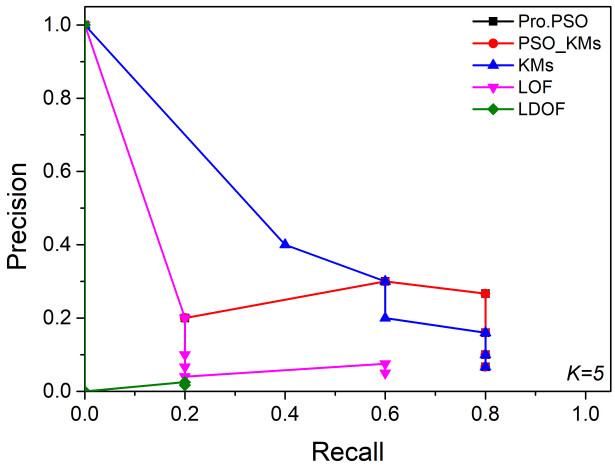

Fig. 6. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=5 for E. coli dataset.

We also considered a bigger dataset i.e., Letter dataset that consist of 55 outliers randomly picked up from 176

Table 15. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF, and proposed PSO for k = 10 of Yeast dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0.1	0.1	1	0.1	0.1	1	0.1	0.1	0	0	0	1	0.1	0.1
15	5	0.33	0.5	4	0.27	0.4	1	0.1	0.06	0	0	0	5	0.33	0.5
20	5	0.25	0.5	5	0.25	0.5	3	0.3	0.15	0	0	0	6	0.3	0.6
25	6	0.24	0.6	6	0.24	0.6	4	0.4	0.16	0	0	0	6	0.24	0.6
40	6	0.15	0.6	6	0.15	0.6	5	0.5	0.12	1	0.1	0.02	6	0.15	0.6

Table 16. F1-score of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k=10 in Yeast dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
5	0	0	0	0	0
10	0.1	0.1	0.1	0	0.1
15	0.4	0.32	0.08	0	0.4
20	0.333	0.333	0.2	0	0.4
25	0.342	0.342	0.228	0	0.342
40	0.24	0.24	0	0.04	0.24

Table 17. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 5, 10 and 30 of Synthetic dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=5	0.1222	0.1139	0.1139	0	0
k=10	0.1872	0.1678	0.1789	0.1003	0.0042
k=30	0.3872	0.3372	0.3372	0.3983	0.1878

Table 18. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF, and proposed PSO for k = 5 of E. coli dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
5	2	0.4	0.4	1	0.2	0.2	1	0.2	0.2	0	0	0	1	0.2	0.2
10	3	0.3	0.6	3	0.3	0.6	1	0.1	0.2	0	0	0	3	0.3	0.6
15	3	0.2	0.6	4	0.27	0.8	1	0.07	0.2	0	0	0	4	0.27	0.8
25	4	0.16	0.8	4	0.16	0.8	1	0.04	0.2	0	0	0	4	0.16	0.8
40	4	0.1	0.8	4	0.1	0.8	3	0.08	0.6	1	0.03	0.2	4	0.1	0.8
60	4	0.07	0.8	4	0.07	0.8	3	0.05	0.6	1	0.02	0.2	4	0.07	0.8

Table 19. F1-score of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k=5 in E.coli dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
5	0.40	0.20	0.20	0	0.20
10	0.40	0.40	0.133	0	0.40
15	0.30	0.40	0.10	0	0.40
25	0.266	0.266	0.066	0	0.266
40	0.177	0.177	0.133	0.044	0.177
60	0.123	0.123	0.092	0.030	0.123

outlier data and added to 1498 normal data, thus the resulting dataset having 1553 data points was used in the experiment. The precision and recall values calculated for all five methods at different threshold values for k=10 is provided in Table 21. It was found that when the threshold value was set at 50 possible outliers, proposed PSO can find out 15 out of 55 outliers, K-means can detect 12 outliers, PSO+K-means can detect 17 outliers, LOF can identify 26 outliers, and LDOF could find out 16 outliers respectively. When we further increase the threshold value up to 100, our proposed PSO could detect 23 out of 55 outliers and LOF could detect a maximum of 33 outliers at k=10 among

Table 20. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 3, 5 and 20 of Ecoli dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=3	0.1089	0.0914	0.155	0.0747	0
k=5	0.1822	0.1822	0.2044	0.0886	0.0069
k=20	0.1989	0.2656	0.2656	0.2322	0.1022

all methods. It was found that a maximum number of 33 outliers was able to detect with LOF method when threshold was set at n=100, whereas our proposed method could detect 23 outliers at the same threshold value. The experiment was repeated for k=30; k=40, and their results are given in Table S7(a) of the supplementary information. The good performance of LOF method particularly in the letter dataset is attributed to distribution of data points. LOF identify the outliers based on the local neighborhood, hence any outlier data laying very close to the dense cluster can be easily identified, which is difficult for global approach methods. Further, F1-score calculated for all five methods are provided in **??** for k=10, and F1-scores of other k-values i.e 30 and 40 are provided in Table S7(b) of the supplementary information. As we see from the Table 22 and Table S7(b), LOF has the highest F1-score as compared to other methods at maximum threshold value for all the k-values. The calculated average precision of all five methods for k=10, 30, and 40 are given in Table 23. As expected, the highest average precision of 0.4277 was achieved for LOF method at k=10, however, the value drastically reduces to 0.2830 as we increase the k-value to 40. While in our proposed method, the average precision was very consistent throughout the k-values. Hence, adopting k=10 in our proposed method will be efficient enough to determine the outliers in larger dataset with a good precision. The PR curves plotted for all five methods for k=10 are provided in Fig. 7, and PR curves for other k-values are provided in figure S7 of the supplementary information.

Further, we have tested our proposed method on biomedical data namely Cardio dataset. Here, there are total 1697 data points, out of which 50 datapoints were outliers and 1647 data points were normal data. The precision and recall values were calculated at k-values of 5, 10 and 30 for all the five methods. Table 24 listed the results for k=5 at different threshold values. Similar results for k=10 and k=30 are provided in Table S8(a) of the supplementary information. In this type of dataset, our proposed PSO enables to detection of 19 out of 50 outliers with a precision of 0.19 at a threshold value of 100. The same number of outliers were also detected with K-means, and PSO+K-means methods with the same precision value. However, LOF and LDOF could detect only 16 and 11 outliers with precision

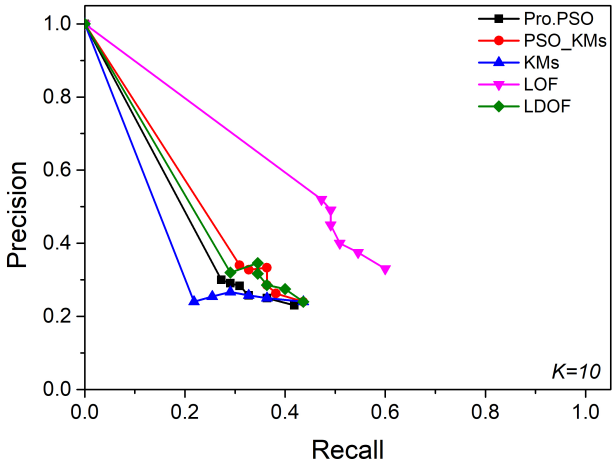

Fig. 7. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=10 for Letter dataset.

of 0.16 and 0.11 respectively. Similar trend was found at k=10. When we increase k=30, PSO+K-means could detect a maximum of 31 outliers, whereas our proposed method and K-means were able to detect 27 outliers only. Both LOF and LDOF can detect at maximum of 22 outliers. A comparison of average precision for all the methods are listed in Table 26 for all the k-values. It was found that our proposed method and PSO+K-means has highest and equal value at k=5 and 10 as compared to other methods. At k=30, the average precision value PSO+K-means (0.3606) was found to be slightly higher than our proposed method (0.3270). Hence, we expected similar performance of both the methods. The corresponding F1-scores are provided in Table 25 for k=5, and Table S8(b) for k=10 and 30 respectively. The PR curves of all the methods for k=5 are shown in Fig. 8. Similarly, the PR curves of k=10 and 30 are provided in figure S8 of the supplementary information. The area under PR curves was found to be equal for our proposed method and PSO+K-means. These results indicate that our proposed method can perform as equally as PSO+K-means method at lower k-values, and performance is higher than the other remaining methods.

Fig. 9 provides a comparative chart of the average precision obtained for all five outlier detection methods dis-

Table 21. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF, and proposed PSO for k = 10 of Letter dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
50	12	0.24	0.22	17	0.34	0.31	26	0.47	0.52	16	0.29	0.32	15	0.3	0.27
55	14	0.25	0.25	18	0.33	0.33	27	0.49	0.49	19	0.34	0.34	16	0.29	0.29
60	16	0.27	0.29	20	0.33	0.36	27	0.49	0.45	19	0.34	0.31	17	0.28	0.31
70	18	0.26	0.33	20	0.29	0.36	28	0.50	0.4	20	0.36	0.28	18	0.26	0.33
80	20	0.25	0.36	21	0.26	0.38	30	0.54	0.37	22	0.4	0.27	20	0.25	0.36
100	24	0.24	0.44	24	0.24	0.44	33	0.6	0.33	24	0.43	0.24	23	0.23	0.42

Table 22. F1-score of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k=10 in Letter dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
50	0.228	0.323	0.495	0.304	0.285
55	0.254	0.327	0.490	0.345	0.290
60	0.278	0.347	0.469	0.330	0.295
70	0.288	0.32	0.448	0.32	0.288
80	0.296	0.311	0.444	0.325	0.296
100	0.309	0.309	0.425	0.309	0.296

Table 23. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 10, 30 and 40 of Letter dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=10	0.2686	0.2981	0.2514	0.4277	0.2971
k=30	0.2341	0.2169	0.1485	0.2897	0.2982
k=40	0.2254	0.1845	0.1199	0.2830	0.2768

Table 24. Comparison of precision and recall for K-means, PSO+K-means, LOF, LDOF, and proposed PSO for k = 5 of Cardio dataset.

	K-means			PSO+K-means			LOF			LDOF			Proposed PSO		
n	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re	n_t	Pr	Re
45	10	0.22	0.2	11	0.24	0.22	12	0.27	0.24	5	0.1	0.1	11	0.24	0.22
50	11	0.22	0.22	12	0.24	0.24	12	0.24	0.24	6	0.12	0.12	12	0.24	0.24
60	13	0.22	0.26	14	0.23	0.28	13	0.22	0.26	8	0.13	0.16	14	0.23	0.28
70	15	0.21	0.3	15	0.21	0.3	14	0.2	0.28	8	0.11	0.16	15	0.21	0.3
80	16	0.2	0.32	16	0.2	0.32	14	0.18	0.28	9	0.11	0.18	16	0.2	0.32
100	19	0.19	0.38	19	0.19	0.38	16	0.16	0.32	11	0.11	0.22	19	0.19	0.38

Table 25. F1-score of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k=5 in Cardio dataset.

n	K-Means	PSO+K-means	LOF	LDOF	Proposed PSO
45	0.210	0.231	0.252	0.105	0.231
50	0.22	0.24	0.24	0.12	0.24
60	0.236	0.254	0.236	0.145	0.254
70	0.25	0.25	0.233	0.133	0.25
80	0.246	0.246	0.215	0.138	0.246
100	0.253	0.253	0.213	0.146	0.253

cussed in this work on eight different datasets. It can be seen from Fig. 9 that the performance and average precision of our proposed method in finding outliers was highest in six different datasets (i.e., Forest Fire(FF), Ionosphere (IS), Wisconsin Breast Diagnostic Cancer (WDBC), Synthetic data (SD), Yeast, Cardio). The average accuracy of proposed PSO is comparable to PSO+K-means in the cardio dataset. In case of E.coli dataset, proposed PSO has equal average precision with PSO+K-means, but there values were slightly lesser than the K-means method. In letter dataset, LOF has the highest average precision value among all methods and performance of our proposed is

Table 26. Comparison of Average Precision for Proposed PSO, PSO+K-means, K-means, LOF, and LDOF for k = 5, 10 and 30 of Cardio dataset.

KNN value	Proposed PSO	PSO+K-means	K-Means	LOF	LDOF
k=5	0.2203	0.2203	0.2105	0.2097	0.1168
k=10	0.2744	0.2745	0.2638	0.2372	0.1514
k=30	0.3270	0.3606	0.3187	0.2335	0.2554

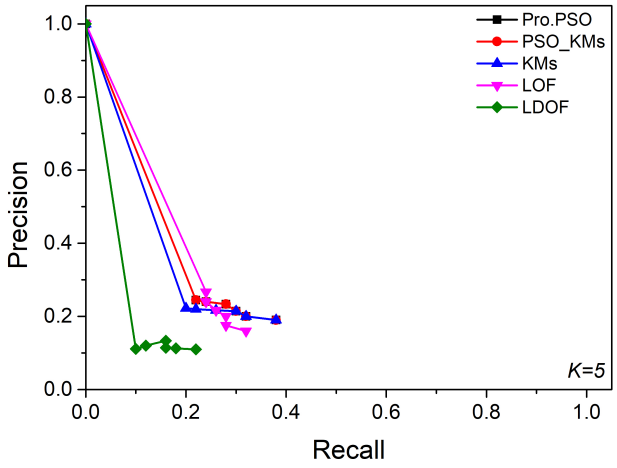

Fig. 8. Precision and Recall curve of Proposed PSO, PSO+K-means, K-means, LOF, and LDOF evaluated with k=5 for Cardio dataset.

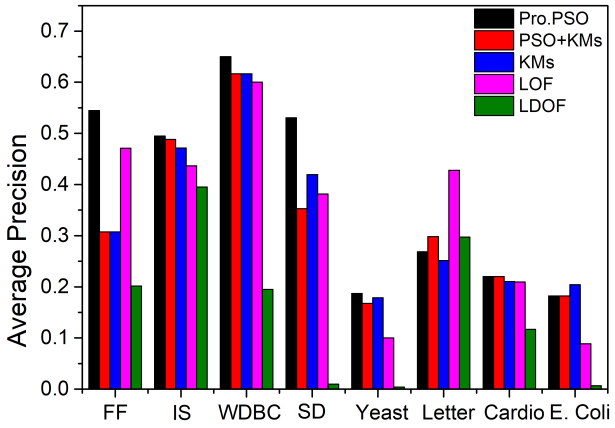

Fig. 9. Comparative average precision plot for proposed PSO, PSO+K-means, K-means, LOF and LDOF for all eight datasets used in the experiment.

slightly lesser. However, as discussed above, our proposed PSO has consistent performance at different k-values, which was not observed in case of LOF method. Thus, our experimental analysis indicate that our proposed PSO method works very well to detect outliers in 6 different types of dataset under investigation as compared to the other methods. We believed that our proposed PSO algorithm could be another promising outlier detection method, particularly in biomedical applications.

7. Conclusion

In this paper, we proposed a simple, efficient and robust outlier detection method by adopting PSO with KNN algorithm. We also adopted data pruning to avoid calculating outlier scores for all elements in the dataset. The data pruning helps to exclude possible inliers as well as reduces the complexity of the problem. In our proposed method, DB Index was employed as a PSO fitness function for cluster optimization and subsequently, the outliers were scored based on the distance of KNN. The performance of our proposed method was compared to PSO+K-means, K-means, LOF and LDOF methods. The analysis of experimental results showed that the proposed PSO method has achieved the highest average precision of outlier detection in six out of eight different types datasets as compared to other methods. Further, the area under PR curve was highest for proposed PSO method in six different type of datasets. Therefore, we believe that our proposed PSO algorithm cloud be a promising technique for identifying outliers that may have potential applications particularly in biomedical data analysis and outlier/anomaly detection problems.

Appendix

7.1. K-Means

K-means represents one of the popular and simple clustering techniques based on unsupervised machine learning algorithms. In this technique, the user needs to randomly initialize the K-number of cluster centers so-called the centriods followed by allocating each point in the dataset to the nearest centroid after calculating the Euclidean distance of each data point to the centroid. Thereafter, each cluster's centroid was updated by calculating the average distance of all data point present within the cluster. Subsequently, each data point is re-assigned to the new centroids. By performing the iterative calculation, this process is repeated until the centroid values are stabilized to determine the optimized centroids [63, 64].

The Euclidean distance (D_{zi}) between x_i data points of n-dimensional space and z^th cluster centroid c_z was

calculated using the equation:

$$D_{zi}^{I-1} = \sqrt{\sum_{j=1}^{n} (x_{ij} - c_{zj}^{(I-1)})^2} \tag{10}$$

where, I is the number of iterations, and n is the dimension of the dataset. The clusters' centroids were recalculated to obtain new cluster centroids using the relation:

$$c_z^I = \frac{\sum_{x_i \in z} x_i}{n_z} \tag{11}$$

where x_i represents data points and n_z is the number of data points in the cluster c_z.

K-means algorithm:

1. Selection and pre-processing of data.
2. Initialization of the number of clusters.
3. Perform the K-means algorithm and give the best centroids.
4. To group the data into clusters by using the best centroids got from K-means.

7.2. Particle Swarm Optimization+K-Means (PSO+K-means)

The model is based on the idea of combining PSO and traditional K-means algorithms, which was reported in the earlier literature [31]. Here, the standard K-means algorithm was initially performed to obtain the initial cluster centers. Thereafter, the initial cluster centers were used as particle positions in the PSO algorithm. Finally, the PSO-generated global best particle's positions were considered as the optimal cluster centers, and subsequently clustering of data points in the clusters was performed. The fitness function used in the reported PSO+K-means algorithm is provided in the following equation:

$$f = (1-\alpha) \times \sum_{i=1}^{n} \sum_{j=1}^{K} \|x_i - c_j\| + a \times \sum_{k,j=1}^{K} \|c_k - c_j\| \rightarrow min \tag{12}$$

where n is the dimensional feature of the dataset, K is the number of clusters, x_i is the data points in cluster c_j, $(1-\alpha)$ is the weight ratios of intra-cluster distance and a weight ratio of inter-cluster distance.

PSO+K-means Algorithm:

1. Selection and pre-processing of data.
2. Initialization of iterations, population size, personal learning coefficient, global learning coefficient, number of clusters, etc.
3. Initialize the position and velocity of particles.
4. Perform the K-means algorithm and give the particles' positions (previous cluster centers).
5. Calculate the fitness function using the above Eq. (12).
6. Calculate velocity then update the position of particles.
7. Update the personal best and global best.
8. Find the best solution or best particle and give the best particle's position which gives the best number of centers.
9. Perform clustering with these centers.

7.3. Local Outlier Factor (LOF)

The Local Outlier Factor (LOF) algorithm is an unsupervised anomaly detection method that computes the local density deviation of a given data point with respect to its neighbors. It considers the samples that have a substantially lower density than their neighbors as outliers. Breunig et al. developed a Local Outlier Factor (LOF) for each object in the data collection, which indicates the degree of outlierness [55]. The outlier factor is local in the sense that it only considers the immediate vicinity of each object, contrary to popular belief. Because the LOF value of an object is determined by comparing its density to that of its neighbors. It really has a higher modeling capability than a distance-based approach, which relies solely on the object's density in a particular major way. Higher the value of LOF, the higher the probability of the point being outlier. Here, LOF is used to compare the results of our proposed technique for the detection of outliers.

7.4. Local Distance-based Outlier Factor (LDOF)

Zhang et al. suggested a local distance-based outlier detection method to locate outliers from the data set. The degree to which an object deviates from its surroundings is determined by its local distance-based outlier factor (LDOF) [56]. A high LDOF score for a point suggests that it is deviating more from its neighbors and is therefore likely to be an outlier. Higher the value of LDOF, the higher the probability of the point being outlier. LDOF is used to compare the results of our proposed technique for the detection of outliers.

Data availability statement

The data generated during the current study are available from the corresponding author on reasonable request.

Acknowledgment

Authors like to thank Prof. Rakesh S. Moirangthem of Department of Physics, Indian Institute of Technology (ISM), Dhanbad, Jharkhand, India for his fruitful suggestions during the preparation of the manuscript.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, (2009) *"Anomaly detection: A survey"* **ACM Computing Surveys *41***(3): DOI: 10.1145/1541880.1541882.

[2] Y. Wang and Y. Li, (2021) *"Outlier detection based on weighted neighbourhood information network for mixed-valued datasets"* **Information Sciences *564***: 396–415. DOI: 10.1016/j.ins.2021.02.045.

[3] H.-P. Kriegel, P. Kröger, and A. Zimek, (2010) *"Outlier detection techniques"* **Tutorial at KDD *10***: 1–76.

[4] H. Wang, M. J. Bah, and M. Hammad, (2019) *"Progress in Outlier Detection Techniques: A Survey"* **IEEE Access *7***: 107964–108000. DOI: 10 . 1109 / ACCESS.2019.2932769.

[5] C. C. Aggarwal. "Supervised outlier detection". In: *Outlier Analysis*. Springer, 2017, 219–248.

[6] B. Diallo, J. Hu, T. Li, G. A. Khan, X. Liang, and Y. Zhao, (2021) *"Deep embedding clustering based on contractive autoencoder"* **Neurocomputing *433***: 96–107. DOI: 10.1016/j.neucom.2020.12.094.

[7] J. Wang, W. Yuan, and D. Cheng, (2015) *"Hybrid genetic-particle swarm algorithm: AN efficient method for fast optimization of atomic clusters"* **Computational and Theoretical Chemistry *1059***: 12–17. DOI: 10 . 1016/j.comptc.2015.02.003.

[8] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, (2015) *"A comprehensive review of swarm optimization algorithms"* **PLoS ONE *10***(5): DOI: 10.1371/journal.pone.0122827.

[9] G. A. Khan, J. Hu, T. Li, B. Diallo, and Y. Zhao, (2022) *"Multi-view low rank sparse representation method for three-way clustering"* **International Journal of Machine Learning and Cybernetics *13***(1): 233–253. DOI: 10.1007/s13042-021-01394-6.

[10] T. Nakane, N. Bold, H. Sun, X. Lu, T. Akashi, and C. Zhang, (2020) *"Application of evolutionary and swarm optimization in computer vision: a literature survey"* **IPSJ Transactions on Computer Vision and Applications *12***(1): DOI: 10.1186/s41074-020-00065-9.

[11] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95-international conference on neural networks*. ***4***. IEEE. 1995, 1942–1948.

[12] R. O. Ogundokun, J. B. Awotunde, P. Sadiku, E. A. Adeniyi, M. Abiodun, and O. I. Dauda. "An Enhanced Intrusion Detection System using Particle Swarm Optimization Feature Extraction Technique". In: ***193***. Cited by: 10; All Open Access, Gold Open Access. 2021, 504–512. DOI: 10.1016/j.procs.2021.10.052.

[13] S. Rana, S. Jasola, and R. Kumar, (2011) *"A review on particle swarm optimization algorithms and their applications to data clustering"* **Artificial Intelligence Review *35***(3): 211–222.

[14] C. Guan, K. K. F. Yuen, and F. Coenen, (2019) *"Particle swarm Optimized Density-based Clustering and Classification: Supervised and unsupervised learning approaches"* **Swarm and Evolutionary Computation *44***: 876–896. DOI: 10.1016/j.swevo.2018.09.008.

[15] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. "Understanding of internal clustering validation measures". In: Cited by: 632. 2010, 911–916. DOI: 10.1109/ICDM.2010.35.

[16] M.-D. Yang, Y.-F. Yang, T.-C. Su, and K.-S. Huang, (2014) *"An efficient fitness function in genetic algorithm classifier for landuse recognition on satellite images"* **The Scientific World Journal *2014***: DOI: 10.1155/2014/264512.

[17] D. L. Davies and D. W. Bouldin, (1979) *"A Cluster Separation Measure"* **IEEE Transactions on Pattern Analysis and Machine Intelligence *PAMI-1***(2): 224–227. DOI: 10.1109/TPAMI.1979.4766909.

[18] M.-D. Yang, Y.-F. Yang, T.-C. Su, and K.-S. Huang, (2014) *"An efficient fitness function in genetic algorithm classifier for landuse recognition on satellite images"* **The Scientific World Journal *2014***: DOI: 10.1155/2014/264512.

[19] A. Asma and B. Sadok. "PSO-based dynamic distributed algorithm for automatic task clustering in a robotic swarm". In: ***159***. Cited by: 10; All Open Access, Gold Open Access. 2019, 1103–1112. DOI: 10.1016/j.procs.2019.09.279.

[20] L. Dey and S. Chakraborty, (2014) *"Canonical pso based-means clustering approach for real datasets"* **International scholarly research notices *2014***:

[21] K. Babaei, Z. Chen, and T. Maul, (2019) *"Detecting point outliers using prune-based outlier factor (plof)"* **arXiv preprint arXiv:1911.01654**:

[22] A. G. Gad, (2022) *"Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review"* **Archives of Computational Methods in Engineering *29***(5): 2531–2561. DOI: 10.1007/s11831-021-09694-4.

[23] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, and S. Ur Rehman, (2014) *"Research on particle swarm optimization based clustering: A systematic review of literature and techniques"* **Swarm and Evolutionary Computation *17***: 1–13. DOI: 10.1016/j.swevo.2014.02.001.

[24] A. A. A. Esmin, R. A. Coelho, and S. Matwin, (2015) *"A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data"* **Artificial Intelligence Review *44***(1): 23–45. DOI: 10.1007/s10462-013-9400-4.

[25] E. O. Merza and N. J. Al-Anber. "A Suggested method for detecting outliers based on a particle swarm optimization algorithm". In: ***1897***. 1. Cited by: 0; All Open Access, Bronze Open Access. 2021. DOI: 10.1088/1742-6596/1897/1/012021.

[26] S. M. H. Bamakan, B. Amiri, M. Mirzabagheri, and Y. Shi. "A New intrusion detection approach using PSO based multiple criteria linear programming". In: ***55***. Cited by: 49; All Open Access, Bronze Open Access. 2015, 231–237. DOI: 10.1016/j.procs.2015.07.040.

[27] K.-W. Wang and S.-J. Qin. "A hybrid approach for anomaly detection using K-means and PSO". In: *2nd International Conference on Electronics, Network and Computer Engineering (ICENCE 2016)*. Atlantis Press. 2016, 821–826.

[28] A. Wahid and A. C. S. Rao, (2019) *"A Distance-Based Outlier Detection Using Particle Swarm Optimization Technique"* **Lecture Notes in Networks and Systems *40***: 633–643. DOI: 10.1007/978-981-13-0586-3_62.

[29] L. Guo, (2020) *"Research on anomaly detection in massive multimedia data transmission network based on improved PSO algorithm"* **IEEE Access *8***: 95368–95377. DOI: 10.1109/ACCESS.2020.2994578.

[30] A. Karami and M. Guerrero-Zapata, (2015) *"A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks"* **Neurocomputing *149***(PC): 1253–1269. DOI: 10.1016/j.neucom.2014.08.070.

[31] R. M. Alguliyev, R. M. Aliguliyev, and F. J. Abdullayeva, (2019) *"PSO+K-means algorithm for anomaly detection in big data"* **Statistics, Optimization and Information Computing *7***(2): 348–359. DOI: 10.19139/soic.v7i2.623.

[32] M. Lotfi Shahreza, D. Moazzami, B. Moshiri, and M. Delavar, (2011) *"Anomaly detection using a self-organizing map and particle swarm optimization"* **Scientia Iranica *18***(6): 1460–1468. DOI: 10.1016/j.scient.2011.08.025.

[33] A. Mekhmoukh and K. Mokrani, (2015) *"Improved Fuzzy C-Means based Particle Swarm Optimization (PSO) initialization and outlier rejection with level set methods for MR brain image segmentation"* **Computer Methods and Programs in Biomedicine *122***(2): 266–281. DOI: 10.1016/j.cmpb.2015.08.001.

[34] A. A. d. M. Meneses, M. D. Machado, and R. Schirru, (2009) *"Particle Swarm Optimization applied to the nuclear reload problem of a Pressurized Water Reactor"* **Progress in Nuclear Energy *51***(2): 319–326. DOI: 10.1016/j.pnucene.2008.07.002.

[35] Y. Zhang, S. Wang, and G. Ji, (2015) *"A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications"* **Mathematical Problems in Engineering *2015***: DOI: 10.1155/2015/931256.

[36] X. Tao, X. Li, W. Chen, T. Liang, Y. Li, J. Guo, and L. Qi, (2021) *"Self-Adaptive two roles hybrid learning strategies-based particle swarm optimization"* **Information Sciences *578***: 457–481. DOI: 10.1016/j.ins.2021.07.008.

[37] Z.-G. Liu, X.-H. Ji, Y. Yang, and H.-T. Cheng, (2021) *"Multi-technique diversity-based particle-swarm optimization"* **Information Sciences *577***: 298–323. DOI: 10.1016/j.ins.2021.07.006.

[38] D. Van Der Merwe and A. Engelbrecht. "Data clustering using particle swarm optimization". In: ***1***. Cited by: 657. 2003, 215–220. DOI: 10.1109/CEC.2003.1299577.

[39] B. Xue, M. Zhang, and W. N. Browne, (2013) *"Particle swarm optimization for feature selection in classification: A multi-objective approach"* **IEEE Transactions on Cybernetics *43***(6): 1656–1671. DOI: 10.1109/TSMCB.2012.2227469.

[40] R. Jamous, H. ALRahhal, and M. El-Darieby, (2021) *"A new ann-particle swarm optimization with center of gravity (ann-psocog) prediction model for the stock market under the effect of covid-19"* **Scientific Programming *2021***:

[41] S. Sarkar, A. Roy, and B. S. Purkayastha, (2013) *"Application of particle swarm optimization in data clustering: A survey"* **International Journal of Computer Applications *65***(25):

[42] L. Zajmi, F. Y. Ahmed, and A. A. Jaharadak, (2018) *"Concepts, Methods, and Performances of Particle Swarm Optimization, Backpropagation, and Neural Networks"* **Applied Computational Intelligence and Soft Computing *2018***: DOI: 10.1155/2018/9547212.

[43] J. C. Bansal, P. K. Singh, and N. R. Pal. *Evolutionary and swarm intelligence algorithms*. ***779***. Springer, 2019.

[44] *Mostapha Kalami Heris Evolutionary Data Clustering in MATLAB*. ttps : / / yarpiz . com / 64 / ypml101 - evolutionary-clustering. accessed June 2021.

[45] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, (2018) *"Efficient kNN classification with different numbers of nearest neighbors"* **IEEE Transactions on Neural Networks and Learning Systems *29***(5): 1774–1785. DOI: 10.1109/TNNLS.2017.2673241.

[46] R. Pamula, J. K. Deka, and S. Nandi. "An outlier detection method based on clustering". In: Cited by: 55. 2011, 253–256. DOI: 10.1109/EAIT.2011.25.

[47] D. L. Davies and D. W. Bouldin, (1979) *"A Cluster Separation Measure"* **IEEE Transactions on Pattern Analysis and Machine Intelligence *PAMI-1***(2): 224–227. DOI: 10.1109/TPAMI.1979.4766909.

[48] P. Cortez and A. de Jesus Raimundo Morais. *A data mining approach to predict forest fires using meteorological data*. http://www3.dsi.uminho.pt/pcortez/fires.pdf. 2007.

[49] D. Dua and C. Graff. *Ionosphere Dataset,UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences*. http: / / archive.ics.uci.edu/ml. 2017.

[50] S. Rayana and L. Akoglu, (2016) *"Less is more: Building selective anomaly ensembles"* **ACM Transactions on Knowledge Discovery from Data *10***(4): DOI: 10.1145/2890508.

[51] N. Fachada, M. A. Figueiredo, V. V. Lopes, R. C. Martins, and A. C. Rosa, (2014) *"Spectrometric differentiation of yeast strains using minimum volume increase and minimum direction change clustering criteria"* **Pattern Recognition Letters *45***(1): 55–61. DOI: 10.1016/j.patrec.2014.03.008.

[52] J. Kools. *6 functions for generating artificial datasets*. https : / / www . mathworks . com / matlabcentral / fileexchange / 41459 - 6 - functions - for - generating - artificial-datasets. accessed on June 2021. 2021.

[53] G. K. Patel, V. K. Dabhi, and H. B. Prajapati, (2017) *"Clustering Using a Combination of Particle Swarm Optimization and K-means"* **Journal of Intelligent Systems *26***(3): 457–469. DOI: 10.1515/jisys-2015-0099.

[54] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, (2002) *"An efficient k-means clustering algorithms: Analysis and implementation"* **IEEE Transactions on Pattern Analysis and Machine Intelligence *24***(7): 881–892. DOI: 10.1109/TPAMI.2002.1017616.

[55] M. M. Breuniq, H.-P. Kriegel, R. T. Ng, and J. Sander, (2000) *"LOF: Identifying density-based local outliers"* **SIGMOD Record (ACM Special Interest Group on Management of Data) *29***(2): 93–104. DOI: 10.1145/335191.335388.

[56] K. Zhang, M. Hutter, and H. Jin, (2009) *"A new local distance-based outlier detection approach for scattered real-world data"* **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) *5476 LNAI***: 813–822. DOI: 10.1007/978-3-642-01307-2_84.

[57] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, (2002) *"An efficient k-means clustering algorithm: Analysis and implementation"* **IEEE transactions on pattern analysis and machine intelligence *24***(7): 881–892.

[58] A. Asuncion and D. Newman. *Forest Fire Dataset, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences*. http://archive.ics.uci.edu/ml. 2007.

[59] W. H. Wolberg, W. N. Street, and O. L. Mangasarian. *Breast cancer Wisconsin (diagnostic) data set, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Science*. http: //archive.ics.uci.edu/ml. 1992.

[60] K. Nakai and M. Kanehisa, (1991) *"Expert system for predicting protein localization sites in gram-negative bacteria"* **Proteins: Structure, Function, and Bioinformatics *11***(2): 95–110.

[61] K. Nakai and M. Kanehisa, (1992) *"A knowledge base for predicting protein localization sites in eukaryotic cells"* **Genomics *14***(4): 897–911. DOI: 10 . 1016 / S0888 - 7543(05)80111-9.

[62] C. C. Aggarwal and S. Sathe, (2015) *"Theoretical foundations and algorithms for outlier ensembles"* **Acm sigkdd explorations newsletter *17***(1): 24–47.

[63] J. Wu and J. Wu, (2012) *"Cluster analysis and K-means clustering: an introduction"* **Advances in K-Means clustering: A data mining thinking**: 1–16.

[64] G. Gan, C. Ma, and J. Wu. *Data clustering: theory, algorithms, and applications*. SIAM, 2020.