Cluster Computing (2021) 24:569–585
https://doi.org/10.1007/s10586-020-03136-9

ODRA: an outlier detection algorithm based on relevant attribute analysis method

Abdul Wahid[1] · Annavarapu Chandra Sekhara Rao[1]

Received: 30 January 2020 / Revised: 12 May 2020 / Accepted: 25 May 2020 / Published online: 13 June 2020

Abstract

Advances in data acquisition have generated an enormous amount of data that captures business, commercial, technological and scientific information. However, some occurrences are rare or unusual, irrespective of a large amount of data available. These rare occurrences in data mining are usually referred to as outliers or anomalies. All these rare occurrences are infrequent. Sometimes it varies from 0.01% to 10% depending on the type of application. In recent years, outlier detection has become important in many applications and has attracted considerable attention among the increasing number of data mining techniques. Focusing on this has resulted in several outlier detection algorithms, mostly based on distance or density. However, each method has its inherent weaknesses. Methods based on distance have problems with local density, and methods based on density have problems with low-density patterns. In this paper, we present a new outlier detection algorithm based on the relevant attribute analysis *(ODRA)* for local outlier detection in a high-dimensional dataset. There are two phases of the proposed algorithm. During the preliminary stage, we present a data reduction method that reduces the data set by pruning irrelevant attributes and data points. In the second phase, we propose an outlier detection method based on *k*-NN kernel density estimation. The experimental results on 15 UCI machine learning repository datasets show the supremacy and effectiveness of our proposed approach over state-of-the-art outlier detection methods.

Keywords Unsupervised outlier detection · Distance-based · Density-based · Data set reduction · Nearest neighbours · Kernel density estimation

1 Introduction

An outlier is an observation that deviates significantly and appears to be inconsistent with the other remaining data sets. The definition of an outlier given by Hawkins [10] is: "an outlier is an object which deviates from others to the extent that it is suspected of being generated by a different mechanism". In data mining, these rare or unusual ocuurances are referred as outliers, anomalies, data-defect, deviants and so on. As outliers present in a dataset have a significant impact on the quality of the data and the results of data mining analysis. As a result, the outlier detection in data mining applications has become an essential pre-processing step. The real-world scenario in which outlier detection has received considerable attention compared to other knowledge discovery issues in data mining is the intrusion detection system, financial fraud detection, video surveillance, detection of medical fraud, health monitoring, and so on. In general, the outliers are typically those data points that are substantially different from the surrounding distribution structures. However, the inconsistent useful or significant information may be provided at some point while nothing useful or meaningful information is present on the other attribute, and these attributes are nearly meaningless or insignificant in outliers detection [18]. In this way, it effectively reduces the power of "disaster estimation" and discovers hidden outliers by searching and removing irrelevant or meaningless attributes that cannot provide any useful information for outlier detection. Outlier detection is intended to detect unusual objects which have substantially different behaviors from those which

✉ Abdul Wahid
awahid.nitp@gmail.com

Annavarapu Chandra Sekhara Rao
acsrao@iitism.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, Jharkhand, India

have been predicted. Focusing on this has resulted in the growth of several outlier detection algorithms, mostly based on distance or density strategies. Except for traditional outlier detection algorithms, current research has focused on the detection of local outliers, i.e., objects that deviate significantly from their neighbourhood. But for each approach, there are inherent weaknesses. Techniques based on distances have a local density issue [5], while density-based methods have a low-density pattern issue [23]. On the other hand, traditional outlier detection algorithms assume that each attribute in the data set has an equal contribution in the direction of outlier detection [15]. However, due to the existence of some irrelevant attributes and samples in the data set, the performance of outlier detection is reduced. It is, therefore, essential to remove these irrelevant attributes and objects from the data set. To overcome the above-mentioned problems, this paper proposes a novel approach referred to as ODRA. The proposed method is influenced by a simple notation that suspicious objects have higher variances and are significantly deviating from their neighbours. The proposed ODRA consists of two steps: at the preliminary stage, it reduces the data set by removing irrelevant attributes and samples from the data set. The detailed description can be found in Sect. 3. In the second step, we proposed a new concept of outlier detection and assigned the outlying degree to each point of the data set. The main highlights and contributions to the paper are summarized as follows:

- A data reduction approach for outlier detection is provided through the use of relevant attribute analysis concept. It reduces the size of the original data set by pruning out irrelevant attributes and data points. The advantage is that the size of the data is significantly reduced, and the performance of the outlier detection algorithm is increased.
- An unsupervised outlier detection algorithm is proposed to address the shortcomings of traditional methods based on distance or density. In this algorithm, a *k*-NN kernel density estimation concept is used to estimate the density at the location of the given data point.
- Our proposed outlier algorithm employed three categories of nearest neighbours, *k*-nearest neighbours (*k*-NN), reverse nearest neighbours (RNN), and shared nearest neighbours (SNN). The advantage is that it makes our system more flexible to model different data patterns.
- The performance of our proposed ODRA algorithm is demonstrated by various experiments on 15 real-world data sets and compared with state-of-the-art outlier detection algorithms.

This paper is organized in the following way. In Sect.2, we present the related work associated with outlier detection algorithms. In Sect.3, we present a detailed description of the data set reduction method, which reduces the size of the data set in order to improve the performance of outlier detection, and a *k*-NN kernel density-based outlier detection algorithm to detect local outliers. In Sect.4, we present an experimental study and an analysis of the results of 15 UCI data sets. Finally, we conclude the full text with future work in Sect.5.

2 Related work

This section provides a brief overview of the various outlier detection algorithms based on the unsupervised method of learning.

2.1 Local outlier detection

In data mining, outliers are a widely discussed problem, and researchers have proposed a variety of outlier detection algorithms because outlier detection is a key component of pattern recognition process. Furthermore, outlier identification have been used in a wide variety of applications. Outlier analysis typically has a wide range, and there are numerous approaches to manage different data types. Based on the strategies adopted, outlier detection approaches can be divided into four broad categories: model-based, cluster-based, distance-based, and density-based. The most popular density-based approach for the identification of local outliers are: LOF [5], COF [22], LoOP [13], INFLO [11], and SimplifiedLOF [21]. In Sect. 4, all these techniques are compared with the proposed one. We have to define the following before describing each of these methods in brief. Let D be a data set, $d_k(p)$ represent a distance between point p and its k^{th} neighbour. We can use other suitable distance metric also such as: Euclidean distance, Mahalanobis distance, etc. The measurement of the distance usually depends on variable types. Let $kNN(p)$ represent the k nearest neighbours of point p, defined as:

$$kNN(p) = q \in D - p : d(p,q) \leq d_k(p) \tag{1}$$

The local reachability density (lrd) of point p can be determined according to the LOF as:

$$lrd(p) = \frac{1}{\sum_{q \in kNN(p)} \frac{Rd_k(p,q)}{|kNN(p)|}} \tag{2}$$

where $Rd_k(p, q)$ is the reachability distance, defined as:

$$Rd_k(p, q) = max\{d_k(q), d(p, q)\} \tag{3}$$

Therefore, the final LOF score can be measured as:

$$LOF_k(p) = \frac{1}{|kNN(p)| \sum_{q \in kNN(p)} \frac{lrd_k(q)}{lrd_k(p)}} \tag{4}$$

Once the LOF score for each point $p \in D$ is calculated, sort it according to their LOF score in decreasing order. It is clear that the higher it's nearest neighbours' reachability density and the lower a point's reachable density, the higher $p's$ LOF score and the respective points are marked as outliers. In the studies of [22], it has been shown that one $'s$ scores, which have a higher LOF score rather than a threshold value, are more accurate to consider as outliers. Later, many LOF algorithm variants have been proposed. One of the variant of LOF is SimplifiedLOF [21], where the reachability distance of the LOF is replaced with kNN distance, resulting a new density estimation of point p as:

$$dens(p) = \frac{1}{d_k(p)} \tag{5}$$

Later, the studies in [13] suggested a robust density estimator based on distance as:

$$LoOP_k(p) = \frac{|kNN(p)|}{\sum d_k(p, q)^2} \tag{6}$$

Tang et al. [22] have developed a new strategy, called COF, with regard to the underlying data patterns. The set-based nearest (SBN) route has been chosen in the COF [22] to obtain a number of nearest neighbours. Furthermore, it was used to calculate the relative density over the average chain distance of a test point. Similar to $LOF_k(p)$, a higher $COF_k(p)$ value indicates that the point p is an outlier. LOF [5] and COF [22]-based approaches identify the outlier by the relative density distribution. Jin et al. [11] have projected a new density-based outlier detection algorithm called INFLO. In this strategy, the relative density is calculated by considering the influence space $IS_k(p)$, which is the combination of k nearest neighbours and reverse nearest neighbours of an object. The $INFLO_k(p)$ is defined as:

$$INFLO_k(p) = \frac{\overline{den}(IS_k(p))}{den(p)} \tag{7}$$

where $\overline{den}(IS_k(p)) = \frac{\sum_{i \in IS_k(p)} den(i)}{|IS_k(p)|}$, and $den(p) = \frac{1}{d_k(p)}$. A new outlier detection algorithm called the Sparse Data Observers (SDO) has been introduced by Vázquez et al. [24]. They are constructing an observer-based data model in this algorithm. An observer is an object situated in a data cluster equidistant from other observers. The outlying degree of an object is computed by measuring the distance from its nearest observers. A cluster-type data set of highly complex structures perform well in the SDO algorithm. Cheng et al. [7] proposed a new ensemble method for detecting outliers in a complex data set. In this method, the authors combined the concept of isolation forest (IF) and the local outlier factor (LOF) for the calculation of the outlying degree of the object. It also reduces the time complexity of the algorithm by pruning out normal data points and generates outlier candidates for the next step. Recently, Jiang et al. [25] have put forward the idea of a gravitational force for outlier detection, in which each point would be treated as an object with both masses as well as local resultant force (LRF) produced by its neighbours. The proposed algorithm uses the LRF change rate difference to detect outliers in data sets. Since most of the conventional outlier detection methods defined in [3, 5, 16] detect outliers by considering all the attributes of a data set. In most of the case, the data points are sparse in a high-dimensional data set. Due to this, one of the problems is to project high-dimensional data into low dimensional subspace to detect the outliers [2, 12, 18], and the second, it has the dimension-exponential complexity [18]. However, recognizing a significant or meaningful subspace is a challenging task [20].To detect outliers in high-dimensional data, the conventional technique for selecting significant subspaces includes two strategies: relevant subspace technique [4, 14, 15], and the sparsity subspace technique [1, 27, 28].

2.2 Relevant subspace-based outlier detection

The primary concept of the relevant subspace approach is to search the corresponding subspace, which consists of an important or useful attribute for detecting outliers in a data set. Some algorithms are based on the concept of a statistical model of local reference data [2, 12, 18], the linear correlation of local reference data [4, 15], and so on. In the sparsity subspace method, it creates a sparse subspace by using a sparse threshold (the user gives sparse threshold value) and projects all the data points in that subspace. It considers those point as an outlier if it belongs to a sparse subspace. To select relevant subspace for outlier detection, Muller et al. [18] have proposed a new method using the Kolmogorov-Smirnov test, in which it recursively explore the subspace from the low to the high dimension. Because of this technique, the complexity to find the relevant subspace is the exponential time of the dimension [19]. Also, the scalability of size for high-dimensional data is terrible, so it can not be adapted for the large data set. Keller et al. [12] designed a new method to find a set of relevant subspaces using the Monte Carlo approach, in which the outlying degree of a point is calculated from the local subset where data point corresponds to each central subspaces. However, the relevant subspace is detected from a global perspective. Moreover, in the concept of principal component analysis (PCA), Kriegel et al. [15] proposed a technique for outlier detection using Mahalanobis distance with the gamma distribution method. The relevant attributes obtained by this technique is an arbitrary linear correlation subspace and have excellent adaptability to linear data. Similarly, Mohamed et al. [4] proposed a new method to detect the relevant subspace using a sparse density matrix, which allows an evaluation of clustering in the corresponding subspace. The sparse subspace is the space in which the density of points is low compared to the average density, and can be measured using the sparse coefficient threshold. Aggarwal and Philip [1] projected an outlier detection technique for high-dimensional data, in which the outliers are determined through the usage of a genetic algorithm where the primary concept is to project the data which is in high dimension, to a low dimensional subspace. The algorithm proposed by Aggarwal et al. improves the effectiveness of outlier detection, but the drawback is that the accuracy and completeness of the outliers are not ensured. To overcome this, Zhu et al. [30] presented a new algorithm to discover outlier in high-dimensional data set that extracts the idea of outlier making use of instance users, followed by looking through a subspace utilizing a modified genetic algorithm. In this section, we have discussed various popular outlier detection algorithms, where they have local density and low-density patterns problem. In comparison to the algorithms mentioned above, we present a novel unsupervised outlier detection algorithm based on dataset reduction method.

3 Methodology

As depicted in Fig. 1, the proposed model takes the raw data as input and outputs the top O outliers. The proposed model mainly comprises two steps -

1. *Dataset reduction* Based on the raw datasets, a dataset reduction technique is applied to prune irrelevant attributes and data points.
2. *Outlier detection* Based on the reduced dataset, an outlier detection algorithm is applied to mine top O points with high outlying degree as the outliers.

3.1 Dataset reduction method

Each data point exists in different forms, which includes sparse cluster and dense cluster patterns. According to the distribution of data, we present a dataset reduction method in this paper. For the sake of simplicity, we show only a 2-dimensional data set with 303 samples in Fig. 2. As shown in the figure, data points in a dense cluster C1 and a sparse cluster C2 are normal, and three data points o_1, o_2, and o_3 are outliers. Traditional outlier detection algorithms need to detect outlying score for each data point, which leads to a significant increase in the time complexity of the algorithm. To improve the efficiency of the algorithm, we present a data reduction method. The objects in the dense cluster (C1) is filtered out through a data reduction method, ensuring that the amount of data entering the next stage (i.e., outlier detection) is less and outliers are among the reduced data set.

As mentioned in the previous section, most of the traditional algorithms for outlier detection become ineffective in detecting outliers in a large data set because they assume that each attribute in a data set has an equal contribution. The presence of irrelevant attributes in a data set reduces the performance of outlier detection. To achieve better performance, we present a relevant attribute selection method. Our goal is to separate the attributes which are relevant to detect an outlier from those that are not. This paper presents an attribute relevance analysis technique in Sect. 3.1.1 to select relevant attributes and data points in a given data set. A similar concept has been adopted in [4, 29].

3.1.1 Relevant attribute analysis

The main aim of the relevant attribute analysis is to remove irrelevant attributes by identifying sparse and dense regions in each attribute of a data set. Where a dense region represents a space in which 1-D point's projected value represents a small cluster structure. A cluster structure indicates a region that has a higher data point density than its neighbours. The dense region includes objects with similar properties that would not be relevant to the detection of outliers. Therefore, it is clear that by detecting the sparse and dense regions in each dimension of a data set, the attributes which are relevant or irrelevant for outlier detection can be distinguished. To detect dense regions in each dimension of a data set, a degree of sparseness for each 1-D point is computed by calculating the variance of its k nearest neighbours (kNN). To calculate the degree of sparseness, we present some notations and definitions. For example, suppose D is a data set containing n data objects in a feature space of dimension m, specifying a set of attributes A denoted as $A = \{A_1, A_2, A_3, \ldots, A_m\}$. Let $D = (p_1, p_2, \ldots, p_n)$ is a set of n data points where $p_i = (p_{i1}, p_{i2}, \ldots, p_{im})$ and each $p_{ij} (i = 1, 2, 3, 4, \ldots, n; j = 1, 2, 3, 4, \ldots, m)$ representing the value of p_i on attribute A_j.

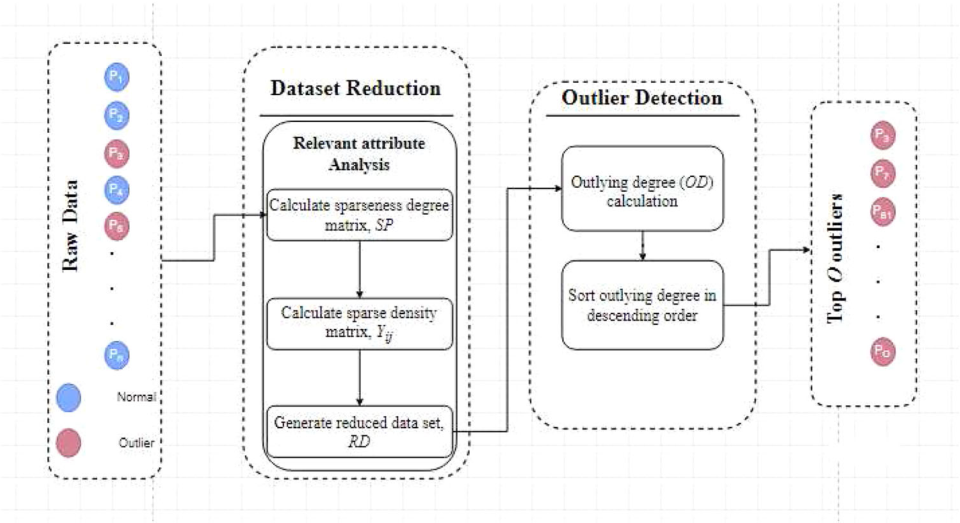

Raw Data
P1
P2
P3
P4
P5
Pn
Normal
Outlier
Dataset Reduction
Relevant attribute Analysis
Calculate sparseness degree matrix, SP
Calculate sparse density matrix, Yij
Generate reduced data set, RD
Outlier Detection
Outlying degree (OD) calculation
Sort outlying degree in descending order
Top O outliers
P3
P7
P81
Po

Fig. 1 The block diagram of proposed ODRA outlier detection algorithm

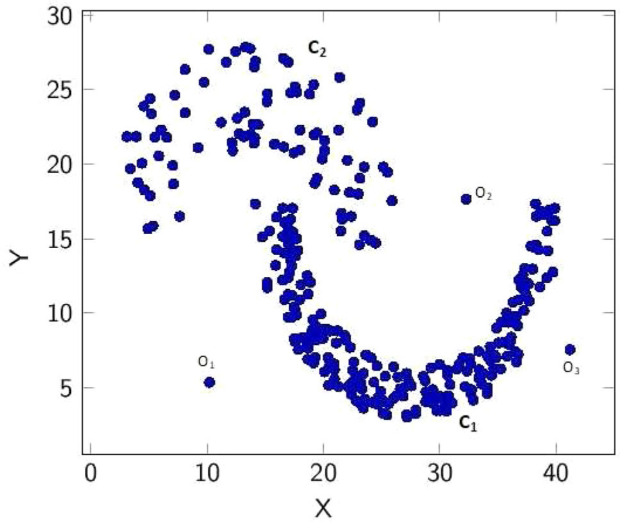


Fig. 2 Distribution of data points

Then we call p_{ij} as a *1-D* point. Given a value of p_{ij} of object p_i on attribute A_j, the sparseness degree is calculated as:

$$Z_{ij} = \sum_{x \in y_i^j(p_{ij})} \frac{(x - c_i^j)^2}{k+1} \tag{8}$$

where $y_i^j(p_{ij})$ is set of p_{ij} and it's k-nearset neighbours. i.e. number of points in $y_i^j(p_{ij}) = k + 1$ or $|y_i^j(p_{ij})| = k + 1$, and $y_i^j(p_{ij}) = nn_k^j(p_{ij})$, where nn_k^j denotes the set of k-nearset neighbours of p_{ij} in dimension A_j, and C_i^j is the center of $y_i^j(p_{ij})$, or we can write C_i^j as:

$$C_i^j = \sum_{x \in y_i^j(p_{ij})} \frac{x}{k+1} \tag{9}$$

Intuitively, a high value of Z_{ij} indicates that p_{ij} comes under the sparse region, while a small Z_{ij} indicates that p_{ij} comes under the dense region. To detect dense regions in each dimension of a data set D, our intention is to find out all sets of p_{ij} , whose sparseness degree (Z_{ij}) is small. Algorithm 1, illustrates the computation of the sparseness degree for each 1-D point in a data set. In general, the calculation of k-nearest neighbours becomes expensive when the number of data points is huge. If we define the value of k to be small, the sparse degree Z_{ij} becomes insignificant. The reason is that very few nearest neighbours may inaccurately derive the sparseness degree of each 1-D point. Instinctively, the value of k must be less than the number of objects in a data set. Since, our data set reduction method is similar to [4], we decide to use $k = \sqrt{n}$, which was configured in [4], whereas k can be chosen according to the application. In Algorithm 1, first, we sort the value of each dimension in ascending order to perform the searching of k-nearest neighbours in 1-D space in an efficient way. Later, we select a kNN set for each 1-D point from its dimensional values and then compute the sparseness degree for each point, according to Eq. (8). At final, we store the sparseness degree of each point in the sparseness degree matrix SP.

Algorithm 1 Compute Sparseness degree

```
Input: Original data set $D$, dimension $m$;
Output: Sparseness degree matrix $SP$;
$n = |D|$;                                  ▷ number of data points in a data set $D$
$k = \sqrt{n}$;                             ▷ number of nearest neighbours
for $(i = 0; i < m; i++)$
    $D_i \leftarrow Sort(D)$;
    for $(j = 0; j < n; j++)$
    $y_i^j \leftarrow Compute(D[i][j])$;
    Compute $Z_{ij}$ from Eq. (8).
    $SP \leftarrow Z_{ij}$
    end for
end for
```

Time complexity: The total time complexity of Algorithm 1 is $O(m \times n)$.

1. The time complexity for sorting n elements is $O(n)$.
2. For finding the kNNs for each objects, the time complexity is $O(n)$.
3. Now, for computing the sparse degree for each point, the time complexity is $O(n)$.

To repeat these three steps for each dimension (i.e. m), the total time complexity is $O(m \times n)$.

(I) *Identify sparse and dense regions:* Here we compute a sparse density matrix Y_{ij} to identify sparse and dense regions in each dimension of a data set. We compare sparseness degree (Z_{ij}) for each 1-*D* point p_{ij} with sparseness degree threshold ϵ. The sparseness degree makes it possible to distinguish sparse and dense regions in each dimension of a data set. Since this is data-dependent, so we cannot use a predetermined value of ϵ to differentiate between sparse and dense regions. Here we use the approach proposed by Bouguessa et al. [4] to select ϵ for the distinction between sparse and dense regions, which is based on the minimum description length principle (MDL). If the sparseness degree of a point is less than ϵ (i.e., $Z_{ij} < \epsilon$), then sparse density value is set to 1, which means p_{ij} belongs to a dense region; otherwise, it is set to 0, indicating that p_{ij} comes under the sparse region. The computational complexity of this algorithm is $O(n \times m)$.

Algorithm 2 Compute Sparse density matrix

```
Input: Sparseness degree matrix $SP$, sparseness degree threshold $\epsilon$;
Output: Sparse density matrix $(Y_{ij})$;
for $(i = 0; i < n; i++)$
        for $(j = 0; j < m; j++)$
          if $(SP[i][j] < \epsilon)$
              $Y[i][j] = 1$;
          else
              $Y[i][j] = 0$;
           end if
       end for
 end for
```

(II) *Generate reduced data set:* Algorithm 3 illustrates how to generate a reduced data set. After getting the sparse density matrix Y_{ij}, Algorithm 3 prunes some attributes and data points which are irrelevant for outlier detection, and then generate a reduced data set *RD*, after comparing with original data set *D*. We use a notation ***IA*** for irrelevant attributes, and ***ID*** for the irrelevant data point in this algorithm.

Algorithm 3 Generate Reduced data set

```
Input: Sparse density matrix ($Y_{ij}$), original data set $D$;
Output: Reduced data set $RD$;
IA $\leftarrow \emptyset$ ;
for ($i = 0; i < m; i++$)
        if ($\sum_{j=0}^{n-1} Y_{ij} = n$)
            $IA \leftarrow IA \cup p_i$
        else
            $RD \leftarrow RD \cup p_i$
        end if
end for
ID $\leftarrow \emptyset$
for ($j = 0; j < n; j++$)
        if ($\sum_{i=0}^{m-1} Y_{ij} = m$)
            $ID \leftarrow ID \cup p_j$
        end if
end for
$RD \leftarrow RD - ID$
```

Time complexity: The total time complexity of this algorithm is $O(n \times m)$

1. The time complexity for column-wise addition of n elements in sparse density matrix (Y_{ij}) is $O(n)$. To repeat the same for m times, the time complexity is $O(m \times n)$.
2. The time complexity for row-wise addition of m elements in sparse density matrix (Y_{ij}) is $O(m)$. To repeat the same for n times, the time complexity is $O(n \times m)$.

Hence, total time complexity for this algorithm is $O(n \times m)$.

3.2 Outlier detection

Outlier detection in data mining applications plays a vital role. An outlier is an occurrence lying an extreme distance from other values in a random sample. A desired outlier detection algorithm does not only generate an output like an inlier or outlier, but should give an outlying degree to every sample of a particular dataset. To compute the outlying degree of an object, the proposed approach initially performs a density estimation. Although in recent years, several density estimation techniques have been suggested. The most common measure of density is the cut-off density, which is determined by the number of items in a r-ball centered on a particular object. However, the parameter r is highly sensitive. The density estimation can vary drastically due to a small variation in r. The kernel density estimator (KDE) is another traditional metric for density estimation, defined as:

$$\rho(p) = \frac{1}{n}\sum_{i=1}^{n} K(\frac{p - p_i}{h}) \tag{10}$$

where $K(.)$ represent a kernel function of width h, fulfilling the following requirements:

$$\int k(x)dx = 1, \int xk(x)dx = 0, \; and \; \int x^2 k(x)dx > 0. \tag{11}$$

KDE is continuous and less susceptible to selecting parameters. However, this tends to give biased estimates of objects in small-sized clusters. Our proposed density estimation is based on k nearest neighbour kernel density (NKD) and takes only nearby objects into account to estimate the density of point p, rather than the whole data set. There are two reasons for this: firstly, the estimated density with whole data can lead to local density loss and local outliers cannot identified with better results. Secondly, using entire datasets for outlying degree computation will increase the computational costs. We use kNN, RNN, and SNN of an object to assess more effectively the density distribution in an object's neighbourhood. The latest research has shown that RNNs can offer useful information on the local data distribution to identify outliers [11]. For an object, the nearest neighbours in kNN(p) should always be k, whereas RNN(p) and SNN(p) may have zero, one or more objects. Given kNN(p), RNN(p), and SNN(p), we create an extended neighbourhood space for an object p by merging in a novel way, represented as:

$$S(p) = kNN(p) \cup RNN(p) \cup SNN(p). \tag{12}$$

Thus, the new local density measure is defined as:

$$\rho(p) = \alpha \sum_{q \in S(p)} exp(\frac{-\delta(p,q)}{\Delta}) \tag{13}$$

where

$$\delta(p,q) = \begin{cases} min_{q\in S(p)} d(p,q), & \text{if } \exists \text{ q such that } \rho(p) < \rho(q) \\ max_{q\in S(p)} d(p,q), & \text{otherwise} \end{cases} \tag{14}$$

$\varDelta$ is the average distance between point p and its k^{th} nearest neighbours, defined as:

$$\varDelta = \frac{1}{n}\sum_{p\in D} d(p, kNN(p)) \tag{15}$$

and α is a controlling parameter ranging from $(0, 1)$.

3.2.1 Outlier detection algorithm

This section presents an algorithm for calculating the outlying degree of an object in its locality. Following the estimation of density at each object, we propose a new outlining approach to evaluating to what extent an object's density differs from its local neighbourhood.

$$OD(p) = \frac{\sum_{q\in S(p)} \rho(q)}{\rho(p)\cdot|S(p)|} \tag{16}$$

The proposed algorithm is the proportion of the average local neighbourhood density to the test point density. The data points with higher density compared to its neighbourhood is very likely to be surrounded by the dense neighbours, indicating that point would not be an outlier, and those with smaller density compared to its neighbours is likely to be an abnormal point. The steps involved in the proposed algorithm are presented in Algorithm 4.

Algorithm 4 Outlier detection algorithm

```
Input: Reduced data set $RD$, $k$, the KNN graph $KNN - G$, number of outliers $O$;
Output:List of top-$O$ otliers.
for each $P \in RD$ do
    $kNN(p) = getOutwardPoints(KNN - G, p)$;
    $RNN(p) = getInwardPoints(KNN - G, p)$;
    $SNN(p) = \phi$;
    for each $q \in kNN(p)$ do
        $RNN(q) = getInwardPoints(KNN - G, q)$;
        $SNN(p) = kNN(p) \cup RNN(p)$;
    end for
end for
for each $p \in RD$ do
    compute outlying degree $(OD)$ for each $p$ according to Eq. (16)
    $OD\_list \leftarrow Sort(OD, 'descending')$;
    Output top-$O$ outliers.
end for
```

3.3 Performance metrics

Since most of the datasets are highly imbalanced, using precision as a performance metric may be inappropriate. Although various performance evaluation metrics have been proposed, they consist of precision, average precision, *Precision@n*, recall, and Area Under the Receiver Operating Characteristic (ROC) curve. In this work, we used four performance metrics: *Precision@n*, average precision (*AP*), F1 score, and area under the ROC curve (*AUC*). *Precision@n*: It is a conventional metric for the quality of an information system's performance, and can be used to measure the overall performance of outlier detection algorithms. It is defined as the proportional of true outliers identified by an algorithm in top n outlier candidates. For a data set D of size N consists of $O \subset D$ outliers and $I \subseteq D$ inliers. *Precision@n* can be calculated as:-

$$Precision@n = \frac{|t_o \in O|rank(t_o) \le n||}{n} \tag{17}$$

If *Precision@n* is used to evaluate the performance of an outlier detection method, it's uncertain that how to select the parameter n. If you set the number of outliers, $n = |O|$, we get the R-precision measure [8]. If the number of outliers $n = |O|$ is minimal compared to N, the value of *Precision@n* may be low and, therefore, not very informative. On the other hand, if $n = |O|$ is significantly high, the value of *Precision@n* may be high because of relatively small proportions of inliers in a data set. A new metric related to information retrieval performance in a wide range of viable choices of n is the average precision (*AP*) [26], where the value of *Precision@n* averaged over the ranks of outlier points $t_o \in |O|$ instead of calculating at a single value of n.

$$AP = \frac{1}{|O|}\sum_{t_o\in O} P@rank(t_o) \tag{18}$$

The F score, also referred to as the F1 score or the F measure, is a common metric that shows the percentage of objects actually defined as the result. The F1 score is calculated as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{19}$$

This score takes both precision and recall into account. The F score achieves the maximum value, i.e., perfect precision and recall with a value of 1. The worst F score, indicating the least precision and least recall, is 0. The most common and powerful performance metric in the anomaly detection literature is based on a curve called the ROC curve [17]. The ROC graph can be determined by evaluation of all possible thresholds, suggesting that the number of samples correctly classified (abnormal scores) known to be true-positive changes with the number of false-positive samples (ordinary or inliers). A graph of the ROC can be summed up with a given metric called Area Under the ROC Curve (AUC). The value of AUC is between 0 and 1. A random algorithm would produce a curve close to the diagonal (AUC score of 0.5), and the best method provides the highest AUC score.

4 Experimental study and results analysis

The aim of this experimental study is to evaluate the efficacy of the proposed algorithm,*ODRA*, which is discussed in this paper. To validate the suitability of *ODRA*, we carried out a series of experiments over 15 real data sets. All the algorithms were implemented in Python programming language, and a performance environment is a machine that includes Intel(R) Core(TM) i7-4770 CPU, 6 GB RAM, and Windows 10 operating system. We compare the performance of our proposed *ODRA* in two aspects. The first part verifies the effectiveness of the data set reduction method in outlier detection. In the second part, by comparing the precision@n, average precision (AP), and AUC (Area Under the ROC Curve) of *ODRA* and some existing algorithms, the efficiency of our proposed algorithm is verified.

4.1 Data set description

We have performed a variety of experiments on 15 real data sets in order to test the effectiveness of proposed outlier detection approach. These data sets were obtained from the UCI Machine Learning Repository[1] and previously used in the literature to evaluate the outlier detection performances. In particular, these all 15 data sets were taken from [6], contain numerous variations of a data set, to evaluate the *ODRA* method in a harsh environment. In [6], two types of data sets were prepared: the data set used in the literature and the semantically significant anomalous values. In this paper, we use both data sets to evaluate the performance of outlier detection techniques that take two pre-processing steps into account: *transformation*, which converts all categorical attributes into numbers using the inverse document frequency (IDF) and *normalization*, which normalizes all attributes between 0 and 1. Table 1 summarizes the details of each data set, showing that the data differs significantly from the others in terms of the number of instances, dimensions, and percentage of outliers.

The *ALOI*, *WPBC*, *Ionosphere*, *PenDigits*, *KDDCup99*, *Lymphography*, *Waveform*, and *WDBC* data sets were used in the literature to detect outliers and the remaining datasets: *Annthyroid*, *Arrhythmia*, *Carditocography*, *Parkinson*, *SpamBase*, *Hepatitis*, and *InternetAds* includes semantically meaningful outliers. For the performance evaluation of classification methods, these 15 datasets have been used initially. To perform outlier detection on these datasets, objects from the minor class are labeled as outliers, and the remaining objects are labeled as normal or inliers. For e.g., the objects in the SpamBase dataset are classified into two classes, namely, spam and non-spam. Since we only consider minor class objects as outliers, this dataset has 133 objects of spam class as outliers and the other objects of non-spam class as inliers. In the same way, the InternetAds dataset, which has 1.9% of the objects as outliers and remaining objects as inliers.

4.2 Effectiveness of dataset reduction method

The criterion for evaluating the effectiveness of the data set reduction method is to filter out normal data points and ensure that outliers are among candidate data sets. That is *Sensitivity*. If the outliers are filtered out due to data set reduction, the data set reduction method is invalid. The calculation of sensitivity is as follows:

$$Sensitivity = \frac{number\ of\ outliers\ in\ candidate\ samples}{number\ of\ outliers\ in\ original\ data\ set} \tag{20}$$

If the sensitivity of the data set reduction method is 100%, the candidate data sets are sensitive to all outliers, which reports that the data set reduction method is effective at this time. We validate the effectiveness of the data set reduction method through several real data sets. Table 2 shows the data sets we selected, the numbers of candidate samples after data set reduction, and the number of outliers in the samples.

By comparing the number of samples and outliers in the original data sets, the number of samples and outliers in the candidate set, we can note that the data reduction method reduces the size of the data set and ensure that outliers still exist in the candidate set. At this time, the normal data points in candidate data sets are still obvious compared

[1] http://www.archive.ics.uci.edu/ml/

Table 1 Description of UCI datasets for experimental analysis

Dataset	# of samples	# of attributes	# of outliers	Outlier. vs inlier.
WDBC	367	30	10	Class 'malignant' to 10 objects (out.)
Waveform	3443	21	100	Class '0' to 100 objects (out.)
Lymphography	148	18	06	Classes 1 and 4 (out.) vs. others (in.)
KDDCup99	48113	40	200	U2R (out.) vs. Normal (inlier)
PenDigits	9868	16	20	Class '4' to 20 objects (out.)
InternetAds	1630	1555	32	Ads vs. other image
Hepatitis	80	19	13	Survival vs. fatal
Ionosphere	351	32	126	Class 'b' (out.) vs. class 'g' (in.)
WPBC	198	33	47	Class 'R' (out.) vs. class 'N' (in.)
SpamBase	2661	57	133	Non-spam vs. spam
Parkinson	50	22	02	Healthy vs. Parkinson
Cardiotocography	1734	21	86	Pathologic, suspect vs. healthy
Arrhythmia	256	259	12	cardiac arrhythmia vs. healthy
ALOI	49534	27	1508	1-1508 objects (out.) vs. others (in.)
Annthyroid	7200	21	534	Hypothyroidism vs. healthy

with outliers, which does not affect the next step of outlier detection. The sensitivity of candidate data sets to outliers is 100%, so the validity of the data set reduction method is validated effectively. The experimental results show that the data set reduction method proposed in this paper is very important. It can reduce the amount of data on the basis of ensuring the accuracy of the algorithm.

4.3 Experimental results

To test the effectiveness of *ODRA*, we run the experiments over 15 real data sets and compared with 5 state-of-the-art outlier detection methods: LOF [5], COF [22], LoOP [13], INFLO [11], and SimplifiedLOF [21]. Since all these comparative methods are nearest neighbour-based and have relatively higher performance ensuing, these are generally used in reality. In Sect. 2, details of all these comparing algorithms have been presented. In our experiment, each outlier detection method was performed for every significant choice of the parameter k between 5 and 100 (or equal to the number of instances if it is less than 100) over each data set. To evaluate the performance of different methods for detecting outliers, firstly we considered the precision at n or $Precision@n$ performance metric, where $n = |O|$. We summarize the $Precision@n$ results for all the experimental data sets in Fig. 3, which shows that our proposed *ODRA* technique has superior performance. Particularly, the *ODRA* technique achieves the best performance for 13 data sets which include *Annthyroid*, *ALOI*, *Arrhythmia*, *Cardiotocography*, *Parkinson*, *SpamBase*, *WPBC*, *Ionosphere*, *InternetAds*, *KDDCup99*, *Lymphography*, *Waveform*, and *WDBC*. The reason is that the *ODRA* technique has removed the dense region of each data set by using a data reduction approach. In particular, if the number of attributes in a data set is very large, the number of attribute reductions will be more. For other two datasets *PenDigits* and *Hepatitis*, the *ODRA* approach indicates competitive performance that is slightly worse than COF. Since $Precision@n$ considers only the number of true outliers among the top-n ranked instances. So, the value of $Precision@n$ can be significantly low for the data set which is having a small proportion of outliers (e.g., *KDDCup99*), or the $Precision@n$ value might be high if the proportion of inliers in a data set is relatively low. A new approach called average precision (AP) has been taken into account to reveal the overall performance of outlier detection techniques. Average precision combines the performance in a wide range of possible choices of n. Instead of evaluating at a single value of n, the value of $Precision@n$ is averaged over the ranks of all outlier points. Fig. 4 shows *AP* for all 15 UCI data sets with different values of k. From the Fig. 4, it is clear that the AP score tends to be low with the stronger imbalance of outliers and inliers in a dataset (e.g., *KDDCup99*, and *PenDigits*). Since all comparing methods, and the proposed *ODRA* approach use k as a model parameter, the F1 scores for each dataset at different k values ranging from 5 to 100 (or equal to the number of instances, if it is less than 100) are summarized as boxplots in Fig. 5. From the results shown in Fig. 5, it can be seen that a single outlier detection method does not outperform other methods all the time. However, the proposed algorithm shows a small variance and high average of F1 score, which implies that the stability and overall detection performance of *ODRA* are high compared to other comparing

Table 2 The number of reduced attributes and samples

Data set	No. of samples	No. of attributes	No. of outliers	No. of candidate samples	No. of candidate attributes	No. of outliers in candidate samples	Sensitivity (%)
WDBC	367	30	10	215	24	10	100
Waveform	3443	21	100	2116	18	100	100
Lymphography	148	18	06	105	15	06	100
KDDCup99	48113	40	200	29741	32	200	100
PenDigits	9868	16	20	6091	12	20	100
InternetAds	1630	1555	32	1139	1291	32	100
Hepatitis	80	19	13	65	16	13	100
Ionosphere	351	32	126	292	25	126	100
WPBC	198	33	47	109	25	47	100
SpamBase	2661	57	133	1887	46	133	100
Parkinson	50	22	02	37	19	02	100
Cardiotocography	1734	21	86	1123	16	86	100
Arrhythmia	256	259	12	117	202	12	100
ALOI	49534	27	1508	31212	16	1508	100
Annthyroid	7200	21	534	4896	17	534	100

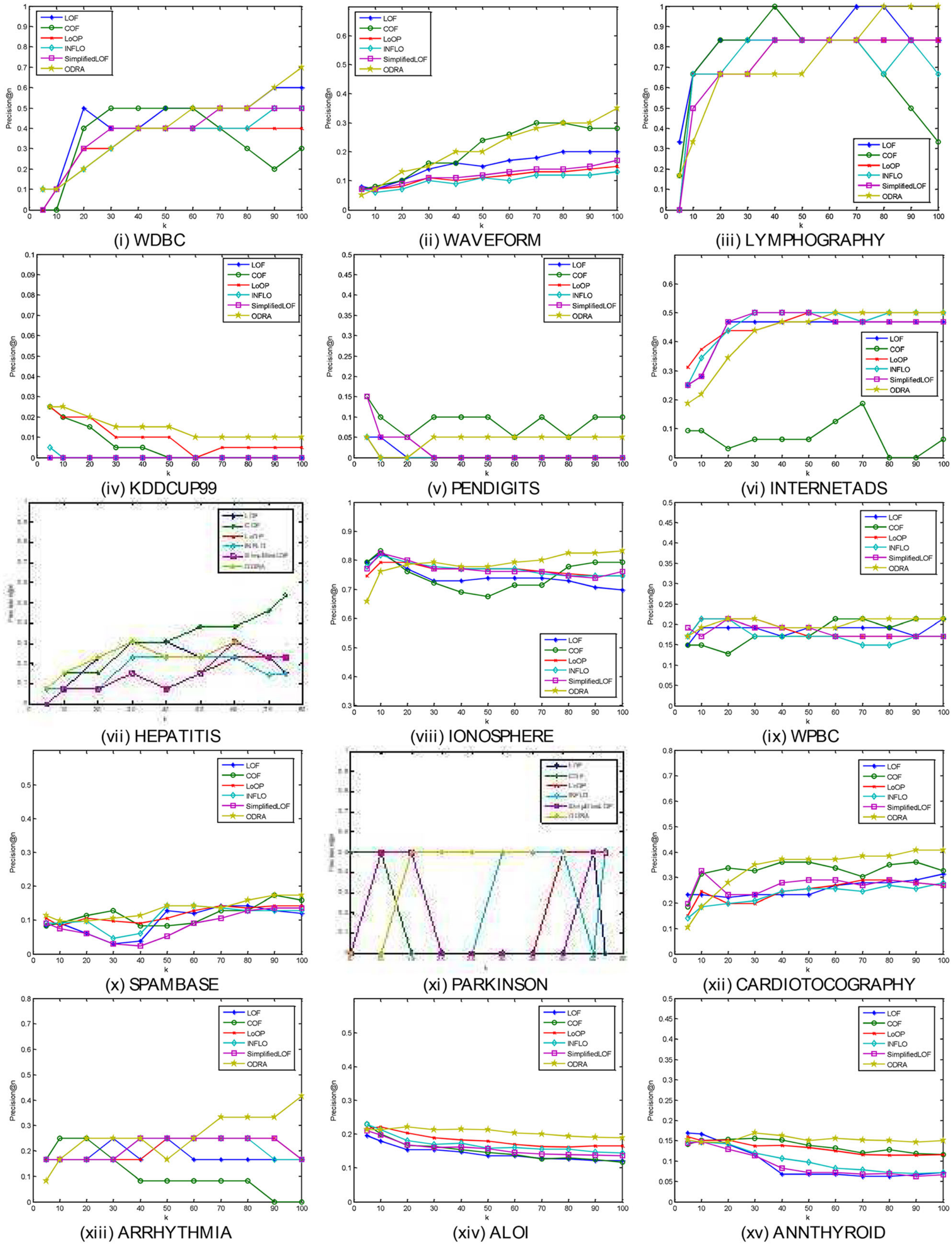
algorithms. Like the precision@n performance metric, the F1 scores achieved by *ODRA* on two datasets: Pendigits and Hepatitis also shows a slightly worse than the COF approach. The most common and powerful performance metric in the anomaly detection literature is based on a curve called the *ROC* curve [17]. The results of the *AUC* were summarized on 15 datasets in Fig. 6, from which it is clear that our proposed approach *ODRA* indicates the supremacy over different comparison algorithms for most of the datasets. For example, in the high-dimensional massive dataset, like *KDDCup99*, the AUC value achieved by *ODRA* is comparatively high (achieved approximately 0.85) compared to other comparing techniques, where the maximum value is around 0.76, which is given by LOF method.

The Friedman's statistical test [9] is used to check whether there is a substantial difference between outlier detection algorithms in a given dataset. It is a two-way evaluation of variance of ranks wherein the null hypothesis (H_0, H_1) defines the performance of all outlier detection algorithms as are the same in terms of given measure against the alternative hypothesis (H_1): represents that at least one approach has substantially different outcomes. If the computed probability is small (the value of p is lower than the selected confidence level), the null hypothesis (H_0) is rejected, indicating that at least two methods differ greatly. In this case, the Nemenyi post-hoc test can be used to show which pairs of methods show these differences. We applied the Friedman test to AUC's best performing scores (i.e., we chose the most powerful k parameter for each data set and method). The Friedman test gave a value of $p = 4.3E - 05$, which shows the significant differences between the outlier detection techniques and, as a result, that rejects the null hypothesis (H_0). The results of the Nemenyi post-hoc test are presented in Table 3.

The symbol "++" or "+" indicates that the method of the column is better than the method of the row with 90% ("++") and 95% ("+") and confidence level, in addition, the symbol '−' and '−−' imply that column method is worse compared to the row method with a confidence level of 95% ('−') and 90% ('−−'). The conclusion can be made from this table is that *ODRA*> LOF> COF> SimplifiedLOF> LoOP> INFLO is true, in which '>' symbol shows "performs better than." We sum up, finally, from all experimental findings discussed above that, compared to other high-dimension dataset approaches, the proposed *ODRA* provides better efficiency for outlier detection.

5 Conclusion

This paper presents an outlier detection algorithm based on relevant attributes analysis to overcome the shortcomings of traditional outlier detection algorithms based on distance or density. Our proposed algorithm *ODRA*, has two main steps. In the first step, our proposed algorithm generates a reduced data set by pruning irrelevant attributes and data points from the original data set. In the second step, we have introduced a novel algorithm for outlier detection. The proposed algorithm is based on the concept of kNN kernel density (NKD) estimation, resulting in a new metric scoring the degree of outlier-ness. In this method, each



(i) WDBC

(ii) WAVEFORM

(iii) LYMPHOGRAPHY

(iv) KDDCUP99

(v) PENDIGITS

(vi) INTERNETADS

(vii) HEPATITIS

(viii) IONOSPHERE

(ix) WPBC

(x) SPAMBASE

(xi) PARKINSON

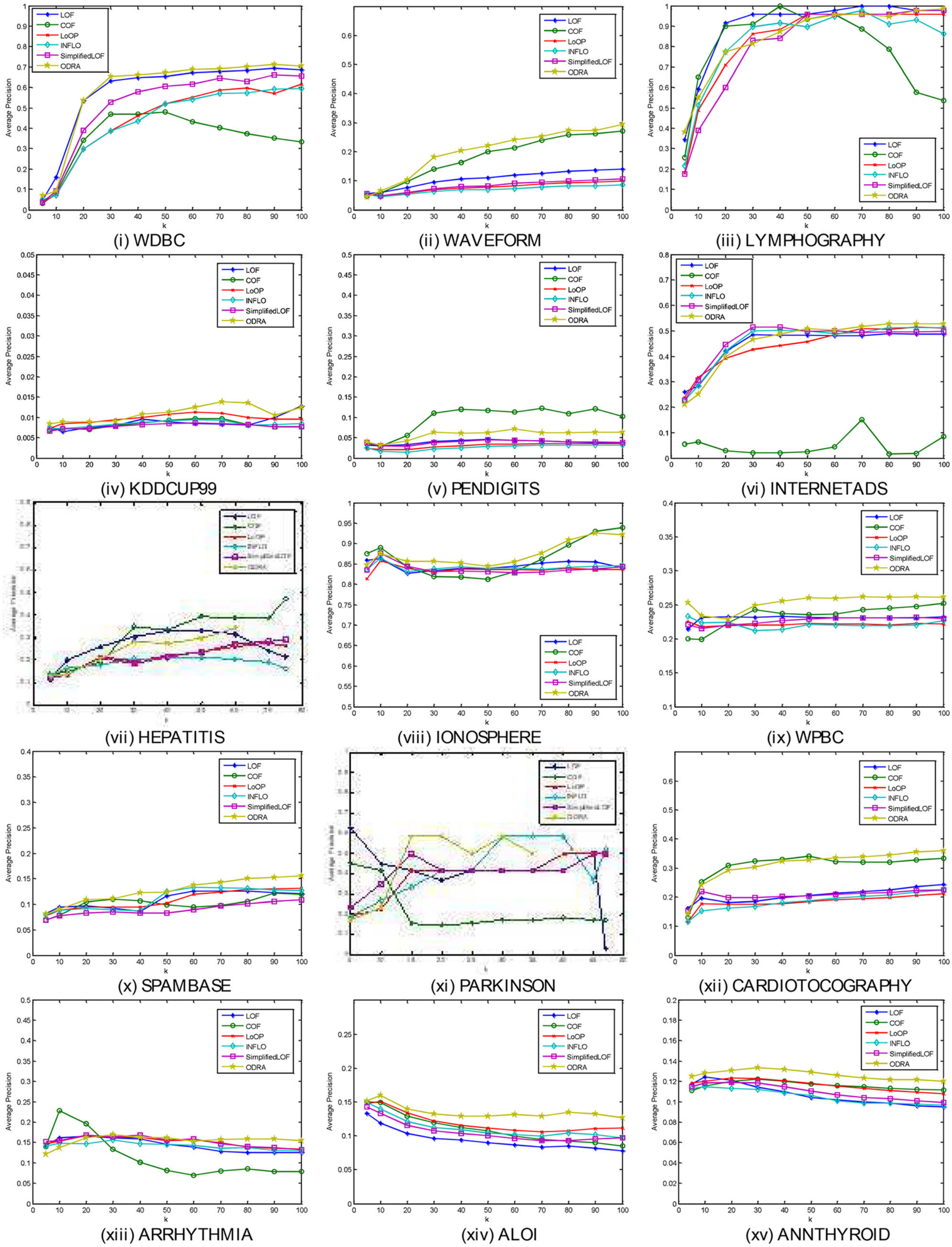
(xii) CARDIOTOCOGRAPHY

(xiii) ARRHYTHMIA

(xiv) ALOI

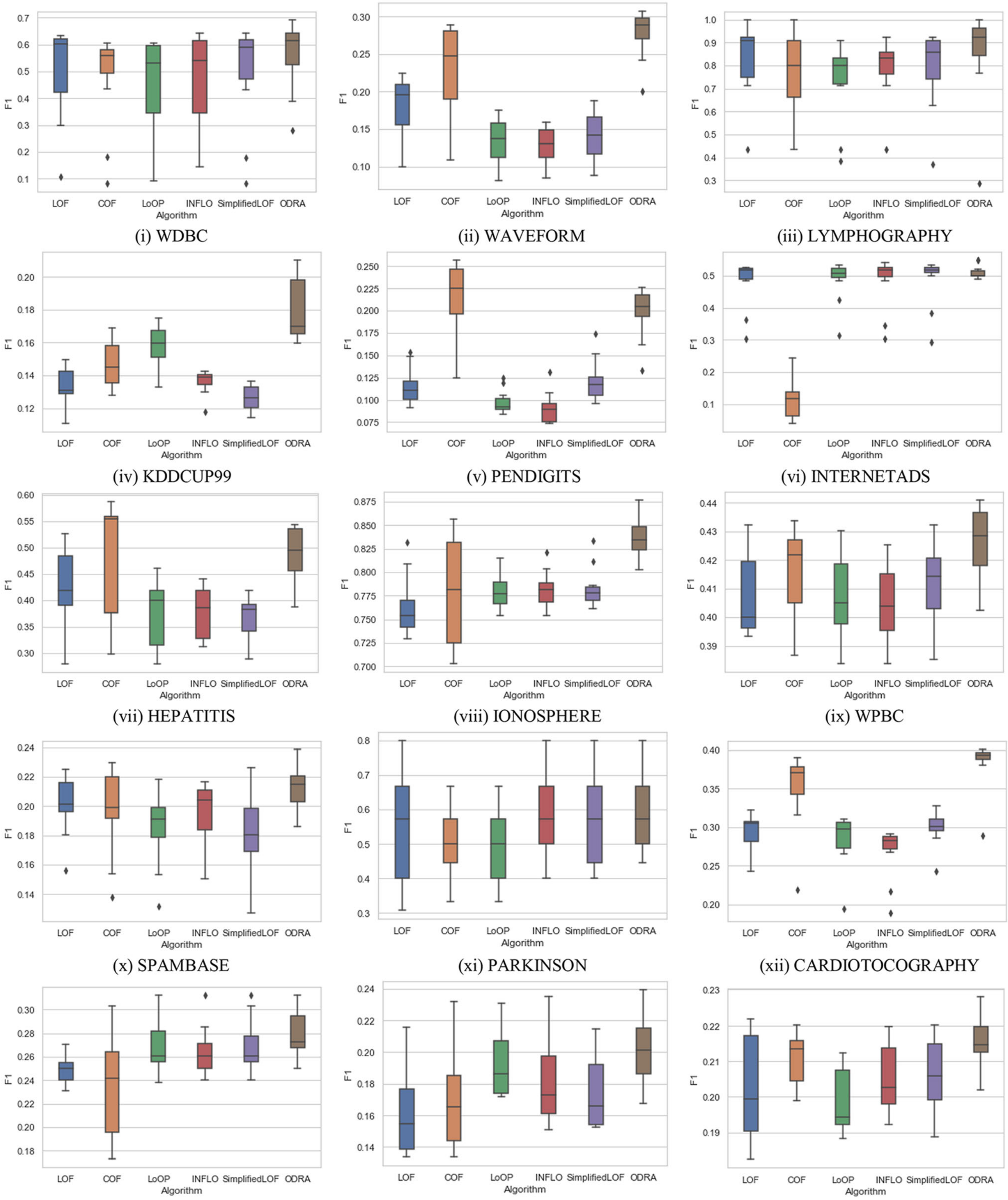
(xv) ANNTHYROID

Fig. 3 Detection performance (Precision@n) of 6 outlier detection methods over 15 UCI datasets



(i) WDBC
(ii) WAVEFORM
(iii) LYMPHOGRAPHY
(iv) KDDCUP99
(v) PENDIGITS
(vi) INTERNETADS
(vii) HEPATITIS
(viii) IONOSPHERE
(ix) WPBC
(x) SPAMBASE
(xi) PARKINSON
(xii) CARDIOTOCOGRAPHY
(xiii) ARRHYTHMIA
(xiv) ALOI
(xv) ANNTHYROID

Fig. 4 Detection performance (Average Precision) of 6 outlier detection methods over 15 UCI datasets



(i) WDBC

(ii) WAVEFORM

(iii) LYMPHOGRAPHY

(iv) KDDCUP99

(v) PENDIGITS

(vi) INTERNETADS

(vii) HEPATITIS

(viii) IONOSPHERE

(ix) WPBC

(x) SPAMBASE

(xi) PARKINSON

(xii) CARDIOTOCOGRAPHY

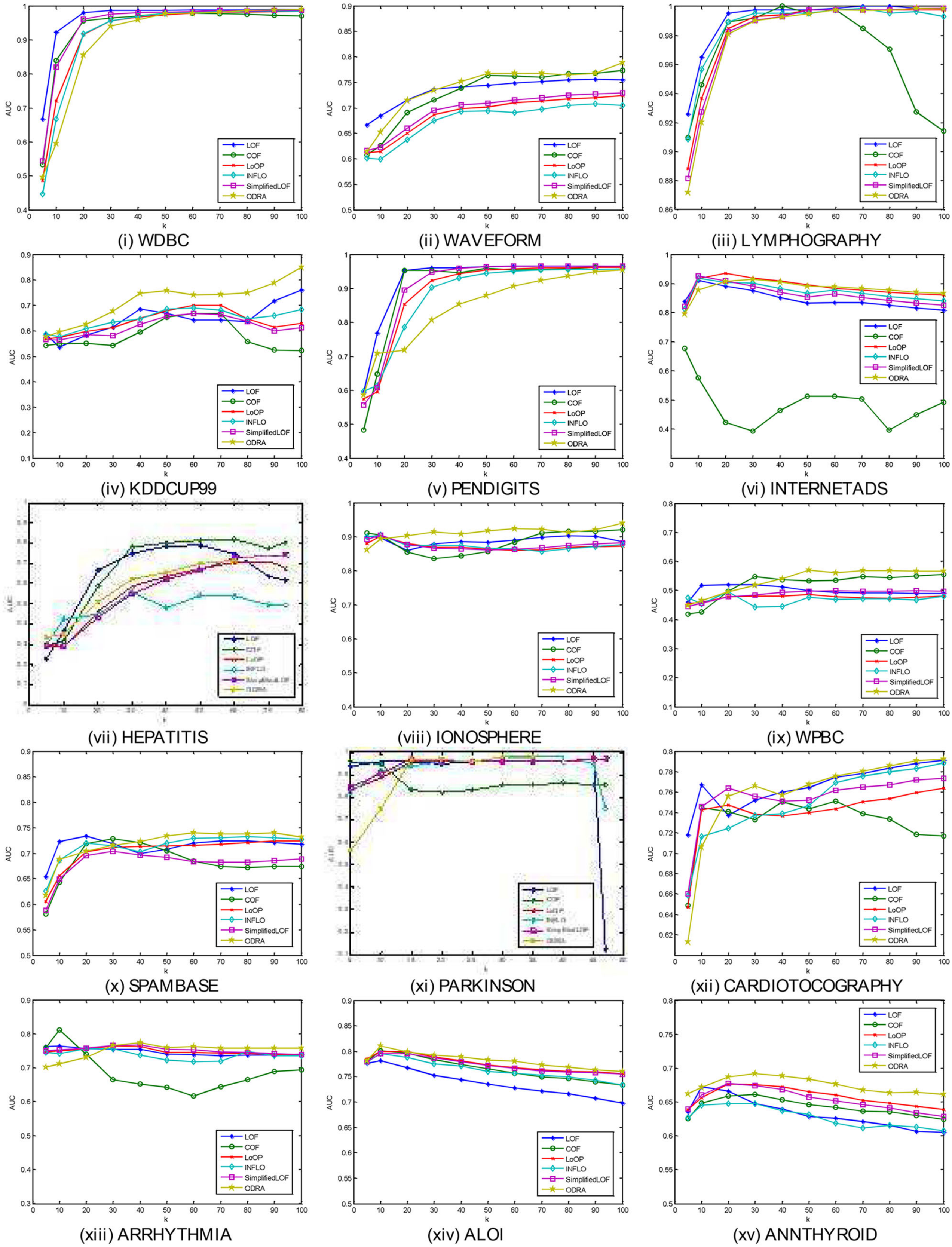
(xiii) ARRHYTHMIA

(xiv) ALOI

(xv) ANNTHYROID

Fig. 5 F1 boxplots of 6 outlier detection methods over 15 UCI datasets

object is assigned a local outlying degree. In particular, the local outlying degree is focused on the notion of local density in which its neighbours give the locality. Further, we use three different types of neighbours: *k*-NN, RNN, and SNN of an object to make our system more flexible in modeling different data patterns. Several experiments were



(i) WDBC

(ii) WAVEFORM

(iii) LYMPHOGRAPHY

(iv) KDDCUP99

(v) PENDIGITS

(vi) INTERNETADS

(vii) HEPATITIS

(viii) IONOSPHERE

(ix) WPBC

(x) SPAMBASE

(xi) PARKINSON

(xii) CARDIOTOCOGRAPHY

(xiii) ARRHYTHMIA

(xiv) ALOI

(xv) ANNTHYROID

Fig. 6 Detection performance (AUC) of 6 outlier detection methods over 15 UCI datasets

Table 3 Statistical differences among the compared algorithms measured by the Nemenyi post-hoc test

	LOF	COF	LoOP	INFLO	SimpliedLOF	*ODRA*
LOF	=		−	−	−	+
COF		=	−	−		+
LoOP	+	+	=			++
INFLO	+	+		=		++
SimpliedLOF	+				=	+
ODRA	−	−	−	−	−	=

conducted over 15 UCI datasets of different characteristics, and five widely used outlier detection algorithms (i.e. LOF, COF, LoOp, INFLO, and Simplified-LOF) are compared to validate the proposed algorithm. Experimental findings demonstrate that our proposed approach performs better than other existing algorithms for outer detection. In the future, proposed work could be expanded to detect outliers in a number of applications such as health care insurance, advertising structures, financial applications, crime and terrorist detection, and so on.

References

1. Aggarwal, C.C., Philip, S.Y.: An effective and efficient algorithm for high-dimensional outlier detection. VLDB J. **14**(2), 211–221 (2005)
2. Aggarwal, C.C., Philip, S.: Outlier detection for high dimensional data. ACM Sigmod. Record. **10**, 37–46 (2001)
3. Barnett, V., Lewis, T., et al.: Outliers in Statistical Data, vol. 3. Wiley, New York (1994)
4. Bouguessa, M., Wang, S.: Mining projected clusters in high-dimensional spaces. IEEE Trans. Knowl. Data Eng. **21**(4), 507–522 (2009)
5. Breunig, M. M., Kriegel, H.-P., Ng, R. T., Sander, J.: Lof: identifying density-based local outliers. In ACM sigmod record, vol.29, pp. 93–104. ACM, (2000)
6. Campos, G.O., Zimek, A., Sander, J., Campello, R.J.G.B., Micenková, B., Schubert, E., Assent, I., Houle, M.E.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining Knowl. Discov. **30**(4), 891–927 (2016)
7. Cheng, Z., Zou, C., Dong, J.: Outlier detection using isolation forest and local outlier factor. In: Proceedings of the conference on research in adaptive and convergent systems, pp. 161–168, (2019)
8. Craswell, N: R-precision, encyclopedia of database systems, (2009)
9. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. **32**(200), 675–701 (1937)
10. Hawkins, D.M.: Identification of Outliers. Springer, New York (1980)
11. Jin, W., Tung, A.K.H., Han, J., Wang, W.: Ranking outliers using symmetric neighborhood relationship. In: Pacific-Asia conference on knowledge discovery and data mining, pp. 577–593. Springer, (2006)
12. Keller, F., Muller, E., Bohm, K.: Hics: high contrast subspaces for density-based outlier ranking. In: Data engineering (ICDE), 2012 IEEE 28th international conference on, pp. 1037–1048. IEEE, (2012)
13. Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A.: Loop: local outlier probabilities. In: Proceedings of the 18th ACM conference on information and knowledge management, pp. 1649–1652. ACM, (2009)
14. Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A.: Outlier detection in axis-parallel subspaces of high dimensional data. In Advances in knowledge discovery and data mining, pp. 831–838, (2009)
15. Kriegel, H.-P., Kroger, P., Schubert, E., Zimek, A.: Outlier detection in arbitrarily oriented subspaces. In: Data mining (ICDM), 2012 IEEE 12th international conference on, pp. 379–388. IEEE, (2012)
16. Kriegel, H.-P., Zimek, A. et al.: Angle-based outlier detection in high-dimensional data. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 444–452. ACM, (2008)
17. Lichman, M.: UCI machine learning repository. irvine, ca: University of california, school of information and computer science. http://archive.ics.uci.edu/ml, (2013)
18. Müller, E., Schiffer, M., Seidl, T..: Statistical selection of relevant subspace projections for outlier ranking. In: Data engineering (ICDE), 2011 IEEE 27th international conference on, pp. 434–445. IEEE, (2011)
19. Pham, N., Pagh, R..: A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 877–885. ACM, (2012)
20. Schubert, E., Zimek, A., Kriegel, H.-P.: Generalized outlier detection with flexible kernel density estimates. In: Proceedings of the 2014 SIAM international conference on data mining, pp. 542–550. SIAM, (2014)
21. Schubert, E., Zimek, A., Kriegel, H.-P.: Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Min. Knowl. Discov. **28**(1), 190–237 (2014)
22. Tang, J., Chen, Z., Fu, A. W.C., Cheung, D.: A robust outlier detection scheme for large data sets. In: In 6th Pacific-Asia conference on knowledge discovery and data mining. Citeseer, (2001)
23. Tang, J., Chen, Z., Fu, A.W.-C., Cheung, D.W.: Enhancing effectiveness of outlier detections for low density patterns. In: Pacific-Asia conference on knowledge discovery and data mining, pp 535–548. Springer, (2002)
24. Vázquez, F.I., Zseby, T., Zimek, A..: Outlier detection based on low density models. In: 2018 IEEE international conference on data mining workshops (ICDMW), pp. 970–979. IEEE, (2018)
25. Xie, J., Xiong, Z., Dai, Q., Wang, X., Zhang, Y.: A local-gravitation-based method for the detection of outliers and boundary points. Knowl. Based Syst. **192**, 105331 (2020)
26. Zhang, E., Zhang, Y..: Average precision. In Encyclopedia of Database Systems, pp. 192–193. Springer, (2009)

27. Zhang, J., Jiang, Y., Chang, K.H., Zhang, S., Cai, J., Hu, L.: A concept lattice based outlier mining method in low-dimensional subspaces. Pattern Recognit. Lett. **30**(15), 1434–1439 (2009)
28. Zhang, J., Zhang, S., Chang, K.H., Qin, X.: An outlier mining algorithm based on constrained concept lattice. Int. J. Syst. Sci. **45**(5), 1170–1179 (2014)
29. Zhao, X., Zhang, J., Qin, X.: Loma: a local outlier mining algorithm based on attribute relevance analysis. Expert Syst. Appl. **84**, 272–280 (2017)
30. Zhu, C., Kitagawa, H., Faloutsos, C..: Example-based robust outlier detection in high dimensional datasets. In: Data mining, fifth IEEE international conference on, pp. 4–pp. IEEE, (2005)

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

Abdul Wahid received his B.Tech degree from Department of Information Technology at Muzaffarpur Institute of Technology Muzaffarpur, Bihar, in 2014 and M.Tech from Department of Computer Science and Engineering at National Institute of Technology Patna, Bihar in 2016. He worked as an Assistant Professor in the Department of Computer Engineering at National Institute of Technology Kurukshetra, Haryana. He is currently a Research Scholar in the Department of Computer Science and Engineering at Indian Institute of Technology (Indian School of Mines) Dhanbad. His research interests include Knowledge Discovery, Data Mining, and Machine Learning.

Annavarapu Chandra Sekhara Rao is currently working as an Assistant Professor in the Department of Computer Science and Engineering at IIT (ISM) Dhanbad. He received his PhD in Computer Science and Engineering from Indian Institute of Technology (Indian School of Mines) Dhanabd, India. He has more than 15 years of teaching experience and published many research papers in reputed journals and conferences. His research interests include Data and Knowledge-based System, Machine Learning, Evolutionary Algorithms, and Bioinformatics.