Computer Science Review 38 (2020) 100306

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review article

A critical overview of outlier detection methods

Abir Smiti *

LARODEC, University of Tunis, Tunisia
Institut Supérieur de Gestion de Tunis, Tunisia

ARTICLE INFO

Article history:
Received 25 May 2020
Received in revised form 6 September 2020
Accepted 17 September 2020
Available online 5 October 2020

Keywords:
Machine learning
Outlier
Noise
Outlier detection

ABSTRACT

One of the opening steps towards obtaining a reasoned analysis is the detection of outlaying observations. Even if outliers are often considered as a miscalculation or noise, they may bring significant information. For that reason, it is important to spot them prior to modeling and analysis. In this paper, we will present a structured and comprehensive review of the research on outlier detection. We have clustered existing methods into different categories based on the underlying approach adopted by each technique. In addition, for each category, we provide a discussion on the advantages and disadvantages of each method. Our paper's purpose is to assist the novice researcher, to produce clear ideas and to facilitate a better understanding of the different directions in which research has been done on this topic.

Contents

* Correspondence to: LARODEC, University of Tunis, Tunisia.
E-mail address: smiti.abir@gmail.com.

https://doi.org/10.1016/j.cosrev.2020.100306
1574-0137/

1. Introduction

A variety of outlier detection techniques have been developed in several research communities. Many of these techniques have been specifically developed for certain application domains, while others are more generic.

If the study of outliers was able to secure credit cards, by identifying suspicious transaction behavior from normal ones and could prevent anomalous network intrusions, just imagine how it could improve the medical field. Medical data analytic will provide better and more effective results and even diseases could be detected when effective outlier detection approaches are developed. These approaches must be robust in the presence of outliers and must provide perfect decision-making towards outliers. In this paper, we will present the state of the art of outlier detection methods. We briefly discuss the differences between noises and outliers. Then we will give a general idea about outlier detection and its influence on data analytic. We have categorized existing methods into different categories based on the underlying approach adopted by each technique. In addition, for each category, we provide a discussion on the advantages and disadvantages of each method.

This paper is formulated in this manner: In the next Section, we present the different outliers' types. In Section 3, we announce a brief resume of outliers methods. Section 4 describes some statistical detection methods and discusses their advantages. Section 5 presents some Distance-based detection methods. Density-based detection methods are presented in Section 6. In Section 7, we are going to show some Cluster-based detection methods. Major pros and cons for each outlier detection approach discussed in this paper are represented in Section 8. The paper concludes in Section 9.

2. Noise VS. outlier

At a first glance, noisy data and outliers may seem alike, however, they really are very different. Hence, before any advancement in our work, we must clear up the ambiguity between the two terms. It would also be important to define all basics which are related to outliers. We will define what an outlier is and view different outliers' types.

2.1. Noise

In data analytic, when talking about noisy data we talk about data from which we have no benefit, i.e., data that only carries meaningless information. A variety of problems appear in the presence of noisy data as machines cannot correctly understand or interpret them, thus, data analytics' results would not be precise enough and unnecessary storage space will be used.

Noisy data can be attributed to various causes:

- **Incorrect data type** (String type entered for a numeric attribute),
- **Erroneous data values** (999 instead of 99 for "age" attribute),
- **Missing values** (Unrecorded data for an attribute).

Attribute 1	Attribute 2
0,25	Red
0,25	Green
1,05	
=	Red
Incorrect type	Missing value

Fig. 1. Noisy data.

For example, Wisconsin Breast cancer (Original) Data set[1] contains missing values i.e. unrecorded attributes . Noisy data holds data analytics back from obtaining satisfying results, therefore their removal would be preferred (See Fig. 1).

2.2. Outlier

Outliers are very different from noisy data, while noises are useless and must be removed, outliers can provide both useless and interesting (exceptional) information. Outliers are defined by the statistician Hawkins [1] as follows:

> *"An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism".*

In other words, outliers are data instances that extremely deviate from well defined norms of a data set or given concepts of expected behavior. In some cases, we would rather remove them as they mislead our analysis, while in other cases they could be very useful and keeping them would be the best solution.

The presence of outliers in data can be attributed to:

- **Measurement or recording errors**,
- **Exceptional but true values**,
- **Mis-reporting**,
- **Sampling error**.

For instance, We may discover that someone suffers from a malignant tumor by identifying an exceptional value in Wisconsin Breast cancer Data set. In general, before any decision-making outlying data must be carefully studied in order to decide weather to keep or delete it. Outlying data that inhibits data analytics should better be removed while those carrying important information should be kept.

2.2.1. Outliers types

In this section different types of outliers are defined. Outliers can be classified into the three types, global outliers, contextual outliers and collective outliers [2].

Global outliers. An outlier is considered as a global outlier, also known as point outliers, when it extremely deviates from well defined norms of a data set or given concepts of expected behavior (see Fig. 2).

[1] WDBC data set is from UCI ML Repository: http://archive.ics.uci.edu/ml.

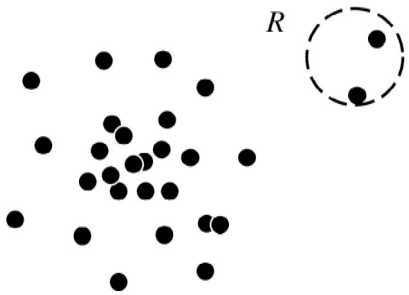


Fig. 2. Global outliers.

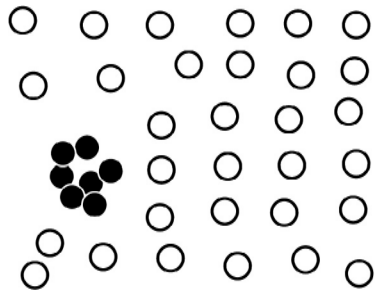


Fig. 3. Collective outliers.

Contextual outliers. When a data object is extremely different in a specific context (not in every context) it is called a contextual outlier. For example, a temperature of 30 °C in Tunis is normal in summer, but it is an outlier for a winter day, it all depends on the context (see Fig. 3).
Each data object can be defined by two attributes:

- Contextual attributes (Date and location in the temperature example)
- Behavioral attributes (Temperature, humidity and pressure in the temperature example)

Collective outliers. When a group of data objects fall extremely far from well defined norms of a data set or given concepts of expected behavior , this collection is known as collective outliers. For instance, a single delayed order is a normal situation for a company that deals with thousands of orders every day, but 100 delayed orders may seem suspicious and this collection of orders forms an outlier.

3. Outlier detection

Outlier detection is the process of detecting outliers in a data set, usually used in the preprocessing phase of data analytic. The detection of potential outliers could be very important for several reasons.

- Data analytic results could be considerably influenced in the presence of outliers.
- When well-studied new information (knowledge) may be discovered.

Outlier detection algorithms are widely adapted in several fields from which we can mention the security, the business and the industry fields. For example, outlier detection can detect suspicious transactions of a credit card or suspicious network attacks can be identified. By implementing outlier detection algorithms in the medical field we can even go further, it can help doctors to detect the early development of cancer tumors and much more.

Outlier detection approaches can be categorized based on different criteria, in our work they will be grouped based on the different techniques they use. Thus we can classify them into four principle methods: statistical-based, distance-based, clustering-based and density-based (See Fig. 4).

4. Statistical detection methods

Statisticians were the first whom observed the presence of outliers as they could easily be identified in statistics. Thus, detecting outlying points became a major challenge and various statistical methods, also known as distribution based, were developed. In these methods a data point is considered as being an outlier if it extremely deviates from a standard distribution. Parametric and non-parametric methods were proposed that need two phases to accomplish the outlier detection process, the training phase and the test phase. In the training phase, all data instances in a data set are trained based on a given statistical model. However, the test phase involves detecting whether a data instance fits into to the model or not, i.e. outlier detection.

4.1. Parametric methods

These methods could be used when we already know the data distribution. Among parametric methods we can mention Gaussian-based and regression-based methods.

4.1.1. GaussIan-based methods

Box-plot [3] and *mean–variance* are the most commonly used techniques in Gaussian-based methods.

Jorma L. et Al. [4] have proposed a method that identifies univariate and multi-variate outliers using boxplots. Boxplots are a simple way of representing the five-number summary which consists five values, the extreme lower(min), the upper extreme (max), the first quartile(Q_1), the median also known as the second quartile (Q_2) and finally the third quartile(Q_3). This technique gives us values for calculating *the range* (Max–Min) and *the inter quartile IQR* ($Q_3 - Q_1$), which can provide us with a boundary for distinguishing normal data from outliers (see Fig. 5).

Example. Consider the following data set: **5, 8, 3, 2, 1, 3, 10**

The values of the data set must be ordered from smallest to biggest in order to apply the five-summary calculations. So the data set will be ordered as follows: **1, 2, 3, 3, 5, 8, 10**.

Obviously, 1 will be the extreme lower and 10 the extreme upper. In our example the data set is composed of 7 values, hence our median will be 3 which is the middle value of the data set. Q_1 and Q_3 are the medians of the data on each side of Q_2, here Q_1 equals 2 and Q_3 equals 8.

- IQR = $Q_3 - Q_1$ = 6
- Outliers will be any points beyond $Q_1 - 1.5 \times IQR = 2 - 9 = -7$ and $Q_3 + 1.5 \times IQR = 8 + 9 = 17$. In our example the data set contains no outliers as no value lies beyond the computed boundary [-7,17].

Authors discovered that boxplots function well with univariate data , however for multivariate data they proposed a more accurate approach that uses the Mahalanobis distance metric.

4.1.2. Regression-based methods

In Fig. 6, figure (a) shows a boxplot of *Y*, which contains one outlier labeled *o*. The same *Y* is regressed on *X*, and the scatterplot, figure (b), shows that *o* cannot be considered as outlier because it lies close to the regression line (see Fig. 6).

For unknown data distributions, regression is used to construct models. A regression model is build in the training phase,

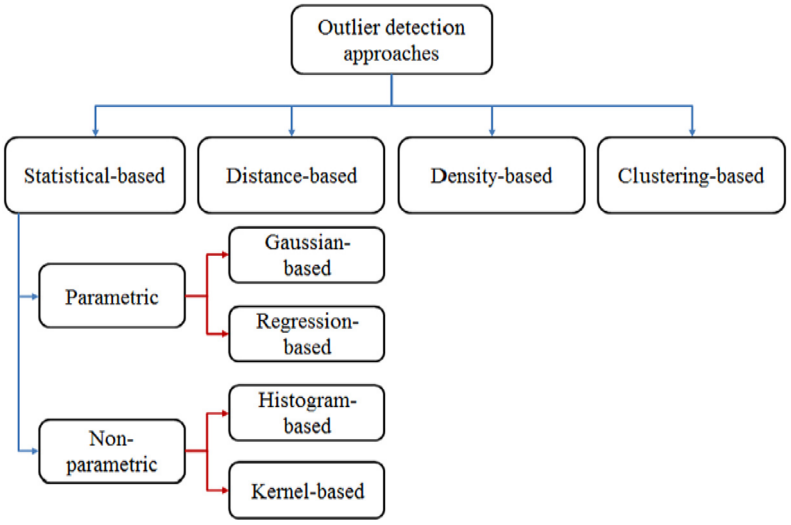


Fig. 4. Outlier detection methods.

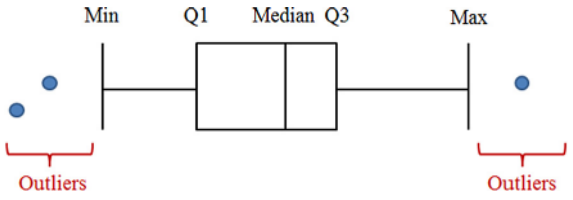


Fig. 5. Boxplot to visualize outlying points.

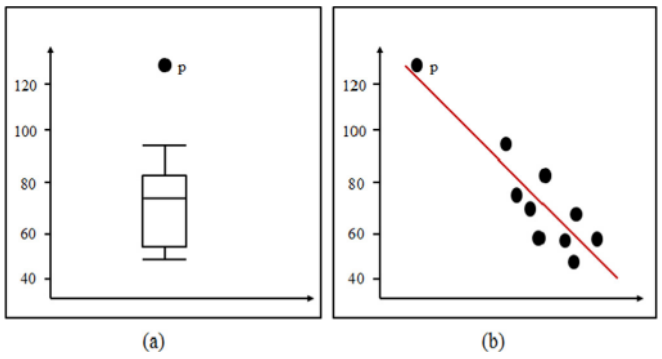


Fig. 6. Regression-based outlier detection.

while the test phase involves testing data objects against the regression model. Data is bivariate in simple regression and can be visualized as a scatterplot, in this case outliers can be easily identified by visual inspection. When data becomes multivariate outlier detection becomes a very difficult task, so several regression methods have been introduced.

In order to handle large data sets, Rousseeuw and Driessen [5] proposed a new regression method called the LTS regression algorithm. Their approach aim to improve the existing LTS (Least Trimmed Square) [6] algorithm which can be defined as follows:

$$minimize_{\beta} \sum_{i=1}^{h} (r^2)_{(i)} \tag{1}$$

Let $(r^2)_{(1)}, (r^2)_{(2)}, \ldots, (r^2)_{(n)}$ be the squared residuals with the respect to their orders. However it's robustness in outlier detection LTS shows low effectiveness in big data sets in terms of computational time.

FAST-LTS was developed in order to cover this drawback. FAST-LTS approach uses different time-saving methods: concentration step (C-step), selective iteration and nested extensions. C-step is responsible for computing the h-subset concentration, a subset with low concentration when it has a low sum of squared residuals. FAST-LTS can reduce computational time by using selective iteration which reduces the number of C-steps. Nested-extensions are used to avoid computing the entire data set.

4.2. Non-parametric methods

Histograms and kernel-based approaches are the most famous non-parametric outlier detection methods.

4.2.1. Histogram-based methods

A histogram is an easy way but also the most used way to visualize statistical data. One of histogram's feature is that it displays the frequency of continuous data, however it cannot visualize categorical data. In histograms, each data class will fall in different ranges. So to decide whether to use histograms or not we must ask ourselves, *"is our data continuous or could it be grouped into ranges?"*.

To construct a histogram data must be separated into intervals, also known as bins, which must be chosen correctly (not to small buckets and not too big) in order to easy see the frequency distribution (see Fig. 7).

Histograms could effectively identify outliers. A histogram-based method was proposed by Markus G. and Andreas D. [7] called the HBOS algorithm. In their proposed algorithm an outlier score for each data instance is computed. Real data sets are composed of features having very different distributions, therefore they used two types of histograms. An univariate histogram is build for each single element, however for numerical elements two types of histograms can be build: static bin-width histograms and dynamic bin-width histograms.

- **Static bin-width histograms-** this type of histograms builds k bins with similar width. In this case to estimate a density, the frequency of objects that fall into this bins is counted.
- **Dynamic bin-width histograms-** To determine a dynamic bin-width values must first be ordered and each bin contains $\frac{N}{k}$ successive values. Let N be the total number of data objects and k the number of bins.

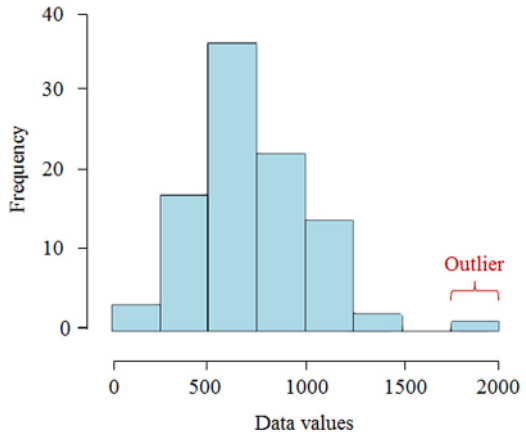


Fig. 7. Histogram to visualize outliers.

Then, all of the histograms are used to calculate an outlier score for each data object. The anomaly score can be calculated as follows [7]:

$$HBOS(p) = \sum_{i=0}^{d} \log\left(\frac{1}{hist_i(p)}\right) \tag{2}$$

Let p be a data instance and the denominator correspond to the height of the bins where the instance is located. HBOS identified global outliers properly, however it failed with local outliers.

4.2.2. Kernel-based methods

As example of this type of methods we will describe the approach proposed by Longin J. L. et Al. [8]. Their approach can be considered as a statistical detection method but also a density-based detection method, as it uses a non parametric kernel in order to estimate the density (also called ground truth density) of data instances. I their method they used the well-known variable width kernel density estimator which can be described in [8]:

$$\widetilde{q}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h(x_i)^{dim}} K\left|\frac{x - x_i}{h(x_i)}\right| \tag{3}$$

where n represents the given data instances of dimensionality dim, K the kernel function and $h(x_i)$ the bandwidth of each data instance x_i. In their method they used the multivariate Gaussian which is defined as follows [8]:

$$k(x) = \frac{1}{(2\pi)^{dim}} \exp\left(-\frac{\|x\|^2}{2}\right) \tag{4}$$

For outlier detection, the algorithm compares densities of each data instance to that of its neighbors. The method succeed in detecting outliers in different data sets which contained outliers of different sizes and in different densities.

4.3. Limits of statistical detection methods

The approaches discussed above showed better results than some existing methods. However, outlier removal in the box-plots approach did not show good performance for all data sets which makes the method data set dependent. The regression-based method reduced the existing LTS approach's computational time even in large data sets but it suffers from the curse of dimensionality. In the case of histograms, it is remarkable that it is very sensitive to the bin sizes, too small bin sizes can lead to false positives and too big bin sizes will produce false negatives. Kernel-based methods show high computational costs used for high dimensionality purposes. However, the nonparametric methods show acceptable performance towards data streams as they can easily deal with continuously arriving data.

Briefly, statistical detection methods show efficient experimental results when probabilistic distribution models are given but suffer from high computational costs when applied in large data sets and the *curse of dimensionality*. Therefore these methods cannot be applied in both large and high dimensional data sets. Another disadvantage of these techniques is that they are not applicable to data sets where the distribution is unknown.

5. Distance-based detection methods

To improve statistical methods limitations several distance-based detection methods have been proposed. In these methods outliers are detected by calculating distances among all data objects based on various distance-related metrics. Afterward, objects that have not enough neighbors are most likely to be outliers. Nearest-neighbor approaches are the most commonly used.

5.1. Solving set approach

This approach was proposed by Fabrizio A. et Al. [9]. The main idea of this method is to use a solving set in order to solve the outlier prediction problem *OPP* and the outlier detection problem *ODP*. When a solving set "S" is defined in a data set "D", distances of objects and their nearest neighbors in the solving set are calculated. Consider the example shown in Fig. 8, figure (a) shows the entire data set, then in (b) a solving set S={a, b, c}, represented by black dots, is defined. The last figure (c) shows neighborhood relationships in which solid arrows represent first and second neighbors of objects in the data set and dashed arrows represent the second neighbors of the solving set objects (See Fig. 8).

In order to compute the ODP solving set three algorithms have been developed, the *SolvingSet algorithm*, the *RobustSolvingSet algorithm* and the *MinimalRobustSolvingSet algorithm*. The principle benefit of this method is that it shows low computational time since only neighborhood distances towards solving set objects are computed instead of the whole data set.

5.2. ABOD approach

The increasing dimensionality of data causes the so called "curse of dimensionality" which means that comparing distances becomes meaningless. Consequently, many distance-based methods become unsuitable. [10] proposed a novel approach called the Angle-Based Outlier Detection ABOD method which still uses distances but also considers the variances of angles of all data objects. The approach observes the variance of angles for each object and computes an outlier factor called CBOF, the further an object is from a cluster the smaller CBOF and the variance of an angle becomes (See Fig. 9).

Fig. 9 object shows how ABOD detects outliers, p is an outlier, we can clearly observe that the angles of inliers are wider than that of outliers. High precision and recall were obtained in experiments which show the effectiveness of ranking outliers.

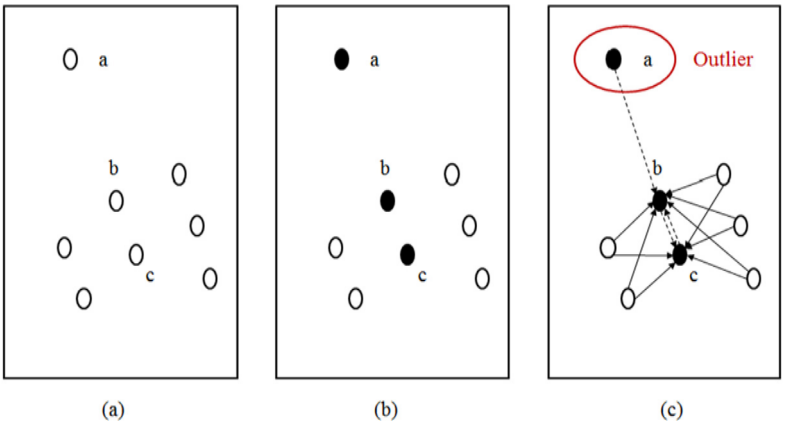


Fig. 8. Solving set method.

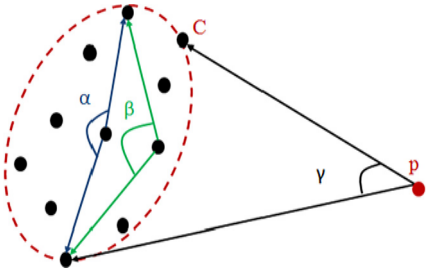


Fig. 9. ABOD.

5.3. LDOF algorithm

Another distance-based algorithm LDOF was introduced by [11]. The algorithm computes LDOF factor on a data instance which indicate how much it deviates from its neighborhood. Data instances obtaining high scores are more likely considered as outliers. LDOF factor is calculated by dividing the KNN distance (see Definition 1) of an object x_p by the KNN inner distance (see Definition 2) of an object x_p

$$LDOF_k(x_p) = \frac{d_{x_p}}{D_{x_p}} \tag{5}$$

Definition 1 (KNN Distance).

"Let N_p be the set of the k-nearest neighbors of object x_p (excluding x_p). The average distance of a data instance x_p to all other data instances in N_p data set is called the KNN distance. To be more formal we can say that dist (x, x') $\geq$ 0 is the distance measure between the two data instances x and x'. The k-nearest neighbors distance of object x_p is defined by [11]:"

$$d_{x_p} = \frac{1}{k} \sum_{x_i \in N_p} dist(x_i, x_i') \tag{6}$$

Definition 2 (KNN Inner Distance).

"Given the k-nearest neighbors set N_p of object x_p, the k-nearest neighbors inner distance of x_p is defined as the average distance among objects in N_p [11]:"

$$D_{x_p} = \frac{1}{k(k-1)} \sum_{x_i, x_i' \in N_p, i \neq i'} dist(x_i, x_i') \tag{7}$$

Top-n outliers are detected, in other words 'n' data instances which have highest outlier score. The approach was experimented on real-world data sets to show its effectiveness. The top-n LDOF algorithm is described as follows:

Algorithm 1 Top-n LDOF algorithm

Input: given data set D, natural numbers n and k.

1. For each object p in D, retrieve k - nearest neighbors;
2. Compute the outlier factor of each object p,
The object with LDOF $\prec LODF_{lb}$ are directly discarded;
3. Rank the objects according to their LDOF scores;
4. **Output:** top-n objects with highest LDOF scores.

5.4. Limits of distance-based detection methods

Distance-based methods may seem effective because of their independence of data distribution and easy implementation. Yet, they still suffer from bad performance in high-dimensional data sets. ABOD was the only discussed technique which overcome this problem. Thus, they cannot handle data streams due to the difficulty of computing distances for data in stream. Another drawback consists in the fact that these methods are computationally expensive when used in multivariate data sets. The solving set method and LDOF may face difficulties when the data distribution is complex. ABOD do not appropriately identify outlier in a low density surrounding area.

6. Density-based detection methods

In density-based techniques density is measured to detect outliers, an outlier is detected when its local density differs from its neighborhood. We can mention LOF and INFLO as well-known approaches which are based on this technique.

6.1. LOF approach

[12] objected the idea of considering an outlier as being a binary property when they proposed LOF the local outlier factor. LOF aims to assign a degree of being an outlier to each data object in a multidimensional data set (See Fig. 10).

The local outlier factor of a data object p is calculated as the ratio of its local density and those of its k-nearest neighbors. The LOF is local in the term that it only considers restricted neighbors of the object. Consider the example of Fig. 10, we can see that the two clusters "C_1'" and "C_2" have different density distributions. Using distance-based methods we cannot identify point "O_2" as

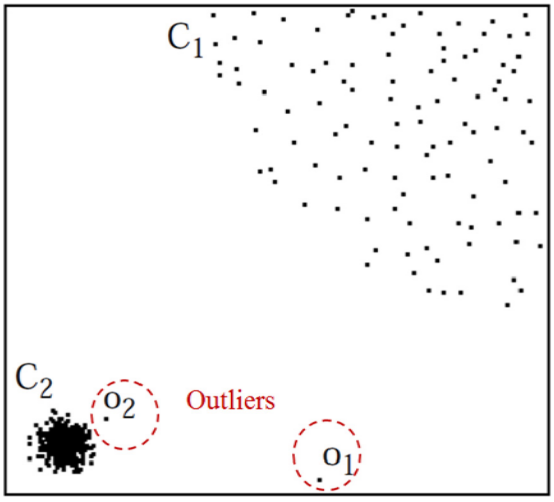


Fig. 10. Local outlier detection based on density.

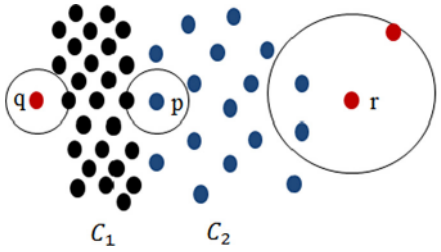


Fig. 11. Motivation for INFLO.

outlier. Here is where LOF outperforms these methods by using the concept of local outliers. LOF demonstrated its importance and quality by identifying more meaningful local outliers than other outlier detection methods.

6.2. INFLO algorithm

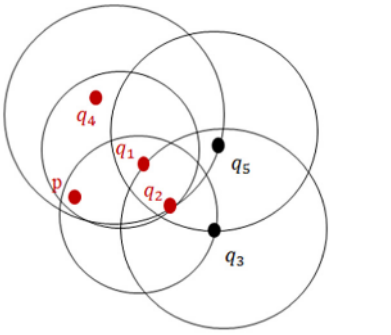
The LOF approach needs well-separated clusters in order to perform well otherwise it suffers from wrong outlier scores, Fig. 11 shows how object p obtains a higher LOF than the two objects r and q which is wrong.

Therefore, [13] proposed another outlier detection approach based on the symmetric neighborhood relationship. In other words, neighbors and reverse neighbors are both considered when estimating its density. The INFLuenced Outlierness degree (INFLO) is affected to each data object in the data set. A high INFLO degree indicates an object may be a promising anomaly candidate while a low score (INFLO $\approx$ 1) means that an object belongs to the core of a cluster.

Fig. 12 shows how INFLO's Influence space (kIS(p)) of an object p includes its KNNs (KNN(p)) and its reverse KNNs (RkNN(p)).

6.3. Limits of density-based detection methods

In general, density-based detection methods show better performance than distance-based detection methods. However they become very computationally expensive when they compute the local density of an object and its neighbors. Hence, they are unsuitable for large data sets and they cannot deal efficiently with data streams. Particularly, LOF experimental results in low density areas was not good. INFLO is sensitive the parameter k which must me choosen appropriately.



$$kIS(p) = kNN(p) \cup RkNN(p) = \{q_1, q_2, q_4\}$$

Fig. 12. Symmetric neighborhood.

7. Cluster-based detection methods

Another important category of outlier detection methods are the cluster-based detection methods. Clustering-based techniques depend on the process of finding different clusters where the objects that does not fit into any cluster are considered as outliers. Several recent approaches are discussed in this section.

7.1. DBSCAN algorithm

As a well-known cluster-based detection method we can mention DBSCAN (Density-Based Spatial Clustering of Applications with Noise), described in algorithm 2, introduced by [14–16]. The principle objective of DBSCAN algorithm is to efficiently cluster scattered data i.e. it can clearly separate clusters of arbitrary forms. Thus, DBSCAN can identify noise in low density areas, i.e., the density of noisy data must be lower than that of the clusters. DBSCAN's effectiveness relies on the three main concepts proposed by Martin E. et Al., the directly density reachability concept (See Definition 1), the density reachability concept (See Definition 2) and finally the density connectivity concept (See Definition 3).

Eps, the maximum radius of the neighborhood, and MinPts, the minimum number of points belonging to Eps neighborhood, are the two required input parameters for DBSCAN.

Algorithm 2 Basic DBSCAN Algorithm

1. Begin
2. Arbitrary select a point p.
3. Retrieve all points that are density-reachable from p w.r.t Eps and MinPts.
4. If p is a core point, form a cluster.
5. If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.
6. Continue the process until each point of the data set has been processed.
7. End

Definition 1 (Directly Density Reachability).

A data object p is considered as being directly-density reachable from a data object q w.r.t Eps and MinPts when:

- $p \in N_{Eps}(q)$
- Core point condition- $|N_{Eps}(q)| \geq MinPts$

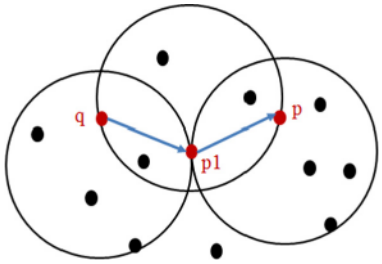


Fig. 13. Density-reachability of object p from object q.

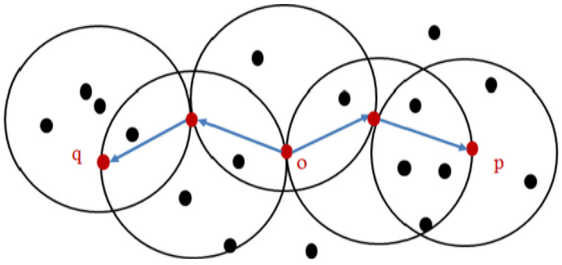


Fig. 14. Density-connectivity of object p to object q.

Definition 2 (Density Reachability).

A point p is density-reachable from a point w.r.t Eps and MinPts if there is a chain of points $p_1, p_2, \ldots, p_n$ such that p_{i+1} is directly-density reachable from p_i [14] (See Fig. 13).

Definition 3 (Density Connectivity).

A data instance p is considered as being density-connected to a data instance q w.r.t Eps and MinPts when data object p and q are both density-reachable from a data object o w.r.t Eps and MinPts (See Fig. 14).

7.2. ODC algorithm

[17] proposed a novel unsupervised approach for outlier detection and clustering improvement by developing the ODC algorithm. The algorithm detects outliers by using a modified version of the well-known K-means (see algorithm 1). As a first step, each object is appointed to the closest centroid. After that the Sum of Squared Error SSE (see definition 1) and Total Sum of Squares SST (see Definition 2) are calculated in order to reduce error and improve clustering quality. To identify outliers ODC calculates the distance between k centroids and all objects and then compares it with the mean of k centroids and all object. If its bigger, the object may be considered as outlier and will be removed. Outlier removal will improve the clustering accuracy.

The algorithm was compared with FindCBLOF and ORC outlier detection methods to confirm its quality. ODC showed better performance in terms of outlier detection and clustering accuracy.

Definition 1 (Sum of Squared Error SSE).

Sum of Squared Error is an important metric for cluster analysis which calculates the Euclidean distance between a data instance and the centroid which it belongs to. This distance is considered as an error and is defined as follows [18]:

$$SSE = \sum_{k=1}^{K} \sum_{\forall x_i \in C_k} \|x_i - \overline{x_k}\|^2 \tag{8}$$

Let C_k be a set of data points belonging to cluster k and $\overline{x_k}$ the clusters' mean.

Definition 2 (Total Sum of Squares SST).

SST is the squared total sum of distances between all the objects in a data set D and the mean. SST is formally defined as [19]:

$$SST = \sum_{\forall x_i \in D} \|x_i - \overline{x_k}\| \tag{9}$$

where $\overline{x_k}$ represents the mean.

The following pseudo code represents the ODC algorithm [17]:

Algorithm 3 ODC algorithm

Input: Data set $D(A_1, A_2, \ldots, A_n)$, number of clusters k, threshold p.
Output: Clustered Data, Outliers and SSE \SST.

1. Choose a value of k.
2. Select k objects randomly and use them as initial set of centroids.
repeat
3. Calculate the distances between k centroids and all the objects in data set D.
4. Calculate the mean distances (M_d) between k centroids and all the objects in data set D.
5. Assign each object to the cluster for which it is nearest centroid and calculate SSE/SST.
for each object $x \in$ data set D **do**
if distance $(x, c_k) \succ p * (M_d)$ **then**
6. Consider x as an outlier and remove from data set D and calculate SSE/SST.
end if
end for
7. Recalculate the centroids.
until objects stop changing clusters

7.3. OF approach

[20] proposed a precise ranking method for anomaly detection based on the idea that outliers are more difficult to identify than inliers (normal objects). Therefore, their approach is based on calculating an Observability Factor OF of each data instance i.e. a degree of inlierness in a data set. The approach examines a part of the data set iteratively, an object that does not fit in the examined part is considered as outlier in the corresponding iteration. This process is repeated and the approach affects an OF score to each object. Those with a low OF score are outlier suspects. Finally all objects are ranked according to their inlierness degree. Results showed that the OF method can effectively identify outliers with stable performances.

The OF algorithm [20] is described in algorithm 5. First a radius ϵ for all samples in the data set is defined as the mean of kth distances between the different objects. Then the algorithm aims to find the nearest neighbors of an object s_i which is denoted as $N_\epsilon(S_i)$. IN_ϵ and OUT_ϵ represent respectively the inlier set and the outlier set. $\delta_r(o_i)$ indicates whether an outlying point must be included to IN_ϵ or OUT_ϵ. Finally, the OF score of each object is computed.

7.4. CLOPD algorithm

[21] proposed the CLOPD (Cluster based Outlying Point Detection) algorithm which also aims to detect anomalies in medical data set. They started with preprocessing raw data to normalize

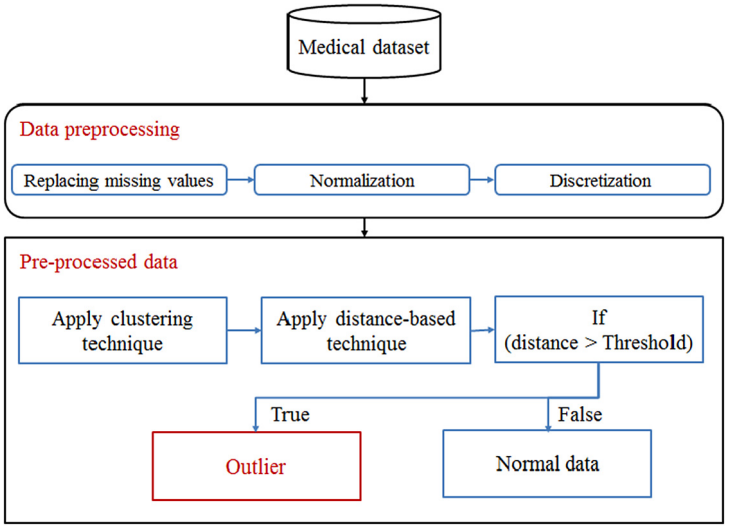


Fig. 15. CLOPD architecture.

Algorithm 4 OF (Observability Factor)

Input: given data set D,number of iterations rep, neighborhood size k.

Output: List of objects ranked by OF values.

$\epsilon \leftarrow$ mean $\{d_k(o_i)|o_i \in D\}$
$M \leftarrow$ F(rep)
for $r \leftarrow 1$ **do**
  rep D_0
  $S \leftarrow$ M random samples from D
  for each $S_i \in S$ **do**
    Find $N_\epsilon(S_i)$
  end for
  $IN_\epsilon \leftarrow \cup_{i=1}^{M} N_\epsilon(S_i)$
  $OUT_\epsilon \leftarrow D - IN_\epsilon$
  if $(O_i \in IN_\epsilon$ **then**
    $\delta_r(o_i) \leftarrow 1$
  end if
  if $(O_i \in OUT_\epsilon$ **then**
    $\delta_r(o_i) \leftarrow 0$
  end if
end for
$OF(o_i) = \sum_{r=1}^{rep} \delta_r(o_i) \backslash rep$

and clean all values in the data set using the z-score normalization method (see Eq. (10)). For outlier detection they used a distance based method, Euclidean distance metric (see Eq. (11)), which help them to find the k nearest neighbors to a point. Outliers are the top M observations that have the largest distance to their kth nearest neighbors (See Fig. 15).

$$Z_i = \frac{x_i j - \overline{x_j}}{S_j} \tag{10}$$

where $\overline{x_j}$ and S_j represent respectively the mean and the standard deviation of an object j.

$$d(x_i, x_j) = \sqrt{\sum_{i=1}^{n} (x_i - x_j)^2} \tag{11}$$

7.5. ROCF algorithm

[22] had identified the top-n parameter problem in almost all of the existing outlier detection algorithms, this means that the algorithm need a parameter n to specify the number of outliers. They developed the ROCF algorithm, described in algorithm 6, which does not need this parameter. The introduced approach is based on the idea that clusters which are smaller than a normal cluster are considered as outliers. Therefore, ROCF uses the Mutual Neighbor Graph (MUNG) to partition the data set, after that a RFOC score is computed to construct the Decision Graph. The Decision Graph helps ROCF detecting not only the outliers but also the outlier clusters, those having a high ROCF score (see definition 1) are outliers.

In Fig. 16, figure (a) displays the entire data set D, figure (b) shows the MUNG graph and finally in figure (c) the Decision graph is constructed. Here C_1 and C_3 are considered as outliers (see Definition 2), C_2 cannot be an outlier because it is the biggest cluster in the data set.

Definition 1 (Relative Outlier Cluster Factor (ROCF)).

The relative outlier cluster factor of cluster C_i, denoted as ROCF (C_i), is defined as follows [22] .

$$ROCF(C_i) = 1 - \exp^{-\frac{n(c_i)}{|c_i|}} = 1 - \exp^{-\frac{|c_i+1|}{|c_i|^2}} \tag{12}$$

Where $i = 1, 2, \ldots, n-1$.

Definition 2 (Outlier Clusters).

Let C_1, C_2, …, C_n be the clusters of the database D. If $ROCF(C_b) = Max\{ROCF(C_i)\}$ and $ROCF(C_b) \succ 0, 1$ then $C_1, C_2, \ldots, C_b$ are outliers clusters [22].

For evaluation, Recall Re and Precision Pr metrics were used. For effective outlier detection results, high values of Re and Pr must be obtained. Huang and all showed RFOC effectiveness by comparing it with the LOF and CBOF algorithms on different data sets. For all experiments RFOC's Recall and Precision were the highest.

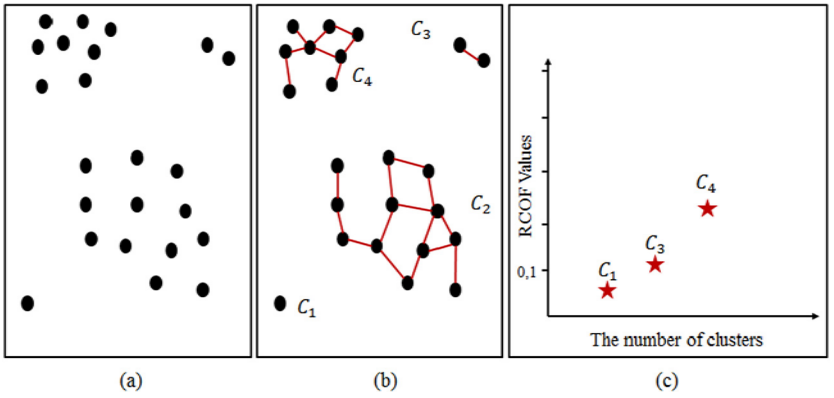


Fig. 16. ROCF algorithm steps.

Table 1
Pros and Cons of outlier detection methods.

Methods	Approach	Pros	Cons
Statistical Methods	Boxplots	+ Easy to use	- Performance is data set dependent
	FAST-LTS	+ Low computational time	- Sensitive to curse of dimensionality
	HBOS	+ Handle data streams	- Cannot detect local outliers
		+ Detect global outliers	- Sensitive to bin sizes
	Kernel-based	+ Detect outliers of different sizes and densities	- Computationally expensive
		+ Handle data streams	
Distance-based Methods	Solving set	+ Low computational time	- Cannot handle data streams
	ABOD	+ Supports High dimensionality	- Sensitive to density
			- Cannot handle low density areas
	LDOF	+ Effective on real data sets	- Cannot handle data streams
			- Cannot handle complex data distribution
Density-based Methods	LOF	+ Detecting meaningful outliers	- Sensitive to density
			- K parameter difficult to select
			- Sensitive to high dimensionality
			- Top-n problem
	INFLO	+ Good outlier ranking	- K parameter difficult to select
			- Finding k is time consuming
Clustering-based methods	DBSCAN	+ Can handle noise	- Sensitive to parameter setting
			- Cannot handle varying densities
	ODC	+ Good clustering accuracy	- No reassign of removed outliers to normal cluster
	OF	+ Precise ranking	- Sensitive to high entropy values
	CLOPD	+ Low computational time	- Difficult to find k parameter
		+ Low error rate	
	RFOC	+ Detect outlier clusters and outliers simultaneously	
		+ Requires few parameters	
		+ No top-n problem	

7.6. Limits of cluster-based detection methods

Experiment results of DBSCAN, ODC, OF, CLOPD and RFOC proved these approaches' efficiency in detecting outliers. DBSCAN suffer from three main drawbacks, it cannot handle the high dimensionality of the data neither it's varying densities. DBSCAN's input parameter must be carefully chosen which makes it parameter setting sensitive. Although ODC clearly improved clustering accuracy it was outperformed by other techniques in terms of run time and overall accuracy. CLOPD showed less computational time and a low error rate. RFOC was compared with the density-based LOF and cluster-based CBOF and obtained the highest recall and precision. In general, cluster-based detection methods does not need any prior knowledge on the data distribution and could be suitable with incremental methods or in other words stream data.

However, cluster-based techniques require too many parameter, thus these parameters are very difficult to be chosen appropriately. RFOC worked on this and only need the k parameter which indicates the number of neighbors. Besides ODC, CLOPD and RFOC use extended versions of the K-means algorithm which itself is sensitive to noisy data and outliers and other clustering algorithms, DBSCAN for example, may show better performance.

8. Summary of outlier detection approaches

Major pros and cons for each outlier detection approach discussed in this paper are represented in Table 1.

9. Conclusion

In this paper, existing outlier detection approaches developed in the past two decades were briefly over-viewed. Statistical-based methods may be effective for given distribution models, but they cannot be applied when this distribution is unknown. Distance-based methods covers this drawback and does not depend on data distribution, however they could be very expensive in multivariate and high dimensional data. Density-based methods are more effective but they remain unsuitable for large data

Algorithm 5 ROCF (C,k)

Input: $C = C_1, C_2, ..., C_n$
Output: $OC = C_1, C_2, ..., C_b$, $b = 0, 1, 2, ..., n-1$

1. for all $C_i \subset C$
a. if size $(C_i) \prec$ k then C_i is marked as outlier cluster and delete C_i from C;
2. for i = 1 : size(C)-1 ;
a. $TL(C_i) = \frac{|C_i+1|}{|C_i|}$
b. $ROCF(C_i) = 1 - \exp^{-\frac{TL(C_i)}{|C_i|}}$
3. Construct and display the Decision Graph.
4. Find the value of b that ROCF(C_b) = max ROCF(C_i) .
a. if ROCF(C_b) $\prec$ 0.1 then b=0;
5. Compute the value of $OR = 1 - \frac{\sum_{i=b+1}^{n}}{|D|}$
6. Output OR and the outlier clusters $OC = C_1, C_2, ..., C_b$.

sets and data streaming. Besides, the cluster-based methods can handle data streams but they still need too many parameters.

Although, many researches have been done in the area of outlier detection, each method still suffer from some drawbacks. New outlier detection methods could be proposed or existing methods could be improved.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Douglas M. Hawkins, Identification of Outliers, vol. 11, Springer, 1980.
[2] Jiawei Han, Jian Pei, Micheline Kamber, Data Mining: Concepts and Techniques, Elsevier, 2011.
[3] Andrew F. Siegel, Charles John Morgan, Statistics and Data Analysis: An Introduction, Wiley, New York, 1988.
[4] Jorma Laurikkala, Martti Juhola, Erna Kentala, N. Lavrac, S. Miksch, B. Kavsek, Informal identification of outliers in medical data, in: Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology, vol. 1, 2000, pp. 20–24.
[5] Peter J. Rousseeuw, Katrien Van Driessen, Computing LTS regression for large data sets, Data. Min. Knowl. Discov. 12 (1) (2006) 29–45.
[6] D.G. Simpson, Introduction to Rousseeuw (1984) least median of squares regression, in: Breakthroughs in Statistics, Springer, 1997, pp. 433–461.
[7] Markus Goldstein, Andreas Dengel, Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm, KI-2012: Poster Demo Track (2012) 59–63.
[8] Longin Jan Latecki, Aleksandar Lazarevic, Dragoljub Pokrajac, Outlier detection with kernel density functions, in: International Workshop on Machine Learning and Data Mining in Pattern Recognition, Springer, 2007, pp. 61–75.
[9] Fabrizio Angiulli, Stefano Basta, Clara Pizzuti, Distance-based detection and prediction of outliers, IEEE Trans. Knowl. Data Eng. 18 (2) (2006) 145–160.
[10] Hans-Peter Kriegel, Arthur Zimek, 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, ACM, 2008, pp. 444–452.
[11] Ke Zhang, Marcus Hutter, Huidong Jin, A new local distance-based outlier detection approach for scattered real-world data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2009, pp. 813–822.
[12] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander, LOF: identifying density-based local outliers, in: ACM Sigmod Record, no. 2, ACM, 2000, pp. 93–104.
[13] Wen Jin, Anthony K.H. Tung, Jiawei Han, Wei Wang, Ranking outliers using symmetric neighborhood relationship, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2006, pp. 577–593.
[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, no. 34, 1996, pp. 226–231.
[15] A. Smiti, Z. Elouedi, DBSCAN-GM: An improved clustering method based on Gaussian means and DBSCAN techniques, in: Proceedings of the 16th International Conference on Intelligent Engineering Systems, IEEE, 2012, pp. 573–578.
[16] A. Smiti, Z. Elouedi, Soft DBSCAN: Improving DBSCAN clustering method using fuzzy set theory, in: Proceedings of the 6th International Conference on Human System Interaction, IEEE, 2013, pp. 380–385.
[17] Mohiuddin Ahmed, Abdun Naser Mahmood, A novel approach for outlier detection and clustering improvement, in: Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on, IEEE, 2013, pp. 577–582.
[18] Lior Rokach, Oded Maimon, Clustering methods, in: Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 321–352.
[19] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis, Clustering validity checking methods: part II, ACM Sigmod Record 31 (3) (2002) 19–27.
[20] Jihyun Ha, Seulgi Seok, Jong-Seok Lee, A precise ranking method for outlier detection, Inform. Sci. 324 (2015) 88–107.
[21] S. Anitha, M. Mary Metilda, A heuristic approach for observing outlying points in diabetes data set, in: Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2017 IEEE International Conference On, IEEE, 2017, pp. 199–202.
[22] Jinlong Huang, Qingsheng Zhu, Lijun Yang, DongDong Cheng, Quanwang Wu, A novel outlier cluster detection algorithm without top-n parameter, Knowl.-Based Syst. 121 (2017) 32–40.