IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 14, 2019, accepted July 29, 2019, date of publication August 2, 2019, date of current version August 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932769

Progress in Outlier Detection Techniques: A Survey

HONGZHI WANG[1,2], MOHAMED JAWARD BAH[1,2], AND MOHAMED HAMMAD[1,3]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[2]Massive Data Computing Laboratory, Harbin Institute of Technology, Harbin 150001, China
[3]Faculty of Computers and Information, Menoufia University, Menoufia 32511, Egypt

Corresponding author: Mohamed Jaward Bah (easybah@yahoo.com)

This work was supported in part by the NSFC under Grant U1509216, Grant U1866602, Grant 61602129, and Grant 61472099, in part by the National Key Research and Development Program of China under Grant 2016YFB1000703, and Microsoft Research Asia.

ABSTRACT Detecting outliers is a significant problem that has been studied in various research and application areas. Researchers continue to design robust schemes to provide solutions to detect outliers efficiently. In this survey, we present a comprehensive and organized review of the progress of outlier detection methods from 2000 to 2019. First, we offer the fundamental concepts of outlier detection and then categorize them into different techniques from diverse outlier detection techniques, such as distance-, clustering-, density-, ensemble-, and learning-based methods. In each category, we introduce some state-of-the-art outlier detection methods and further discuss them in detail in terms of their performance. Second, we delineate their pros, cons, and challenges to provide researchers with a concise overview of each technique and recommend solutions and possible research directions. This paper gives current progress of outlier detection techniques and provides a better understanding of the different outlier detection methods. The open research issues and challenges at the end will provide researchers with a clear path for the future of outlier detection methods.

INDEX TERMS Outlier detection, distance-based, clustering-based, density-based, ensemble-based.

I. INTRODUCTION

Outlier detection remains to be an essential and extensive research branch in data mining due to its widespread use in a wide range of applications. By identifying outliers, researchers can obtain vital knowledge which assists in making better decisions about data. Also, detecting outliers translates to significant actionable information in a wide variety of applications such as fraud detection [1], [2], intrusion detection in cybersecurity [3], and health diagnosis [4]. Despite the ambiguity in providing a clear definition, an outlier is generally considered a data point which is significantly different from other data points or which does not conform to the expected normal pattern of the phenomenon it represents.

Outlier detection techniques strive to solve the problem of discovering patterns that do not adapt to expected behaviors. Consider a scenario where we would want to define the usual behavior and the normal region. This scenario can be complicated because of:

- inaccurate boundaries between the outlier and normal behavior
- the high possibility of the normal behavior to continue to evolve and perhaps it might not be a correct representation in the future
- different applications and conflicting notion make it hard to apply techniques developed in one field to another
- noise in the data which mimics real outliers and therefore makes is challenging to distinguish and remove them.

Although outlier detection faces some challenges, several outlier detection techniques have been proposed that use different methodologies and algorithms to address these issues [5]. Some of the commonly encountered difficulties related to the nature of the input data, outlier type, data labels, accuracy, and computational complexity in terms of the CPU time and memory consumption [6]–[9]. Researchers continue to find better solutions to address these challenges, together with problems associated with detecting outliers efficiently in distributed data streams [10], RFID reading streams [11], large multidimensional data [12], [13], wireless sensor

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammed Nabil El Korso.

TABLE 1. The different categories covered by our survey and other related survey.

Paper & Year	1	2	3	4	5	6	7	8	9	10	11	12	13
Barnett et al. [39] 1994	X												
Hodge et al. [5] 2004	X	X	X	X		X						X	
Walfish et al. [40] 2006	X												
Patcha et al. [41] 2007	X	X		X		X							
Chandola et al. [22] 2009	X	X	X	X									
Hadi et al. [29] 2009	X	X	X										
Gogoi et al. [33] 2011	X	X	X	X		X		X					
Zhang [26] 2013	X	X	X	X					X				
Gupta et al. [31] 2014									X				
Akoglu et al. [34] 2014							X				X		
Ranshous et al. [23] 2015								X					
Aggarwal [28] 2016					X	X			X		X	X	
Kwon et al. [30] 2017						X							
Chalapathy et al. [32] 2019						X							
Ours	X	X	X	X	X	X	X	X	X	X	X	X	X

1. Statistics-based 2. Distance-based 3. Density-based 4. Clustering-Based 5. Ensemble-Based 6. Learning-Based 7. Graph-based 8. Network 9. Data Streams 10. Tools 11. Datasets 12. Applications 13. References later than 2016.

networks [14], efficient trajectories [15], and in data quality and cleaning [16].

For example, consider the challenges present in large multidimensional data, in which, whether the data is relatively large or extremely large, it always contains some outliers. In most cases, as the data increase in size, the number of outliers also increases [17]. Therefore, with a large volume of data, it is essential to design scalable outlier detection techniques to handle large datasets (Volume). As data increase in size, this proportionally influences the computational cost, rendering the process slow and expensive. It is of great importance that these outliers are detected in a timely manner, to minimize dirty data, prevent data infection, and for the data to provide a well-timed value (Velocity and Value). In another case, when varieties of data are present and some of which are structured, mixed-valued, semi-structured and unstructured data (Variety); computing outliers of this nature can be daunting and complicated. Other areas that are confronted with challenges include in application areas such as mobile social networks, security surveillance [18], [239], trajectory streams [19], and traffic management [20], [21]. These areas demand constant discovery of abnormal objects to deliver crucial information promptly. Many other outlier detection areas share similar, and new challenges, and they will be referred to in subsequent sections of this paper.

As a result of the inherent importance of outlier detection in various areas, considerable research efforts in the survey of outlier detection (OD) methods have been made [22]–[34]. Despite the increasing number of reviews in outlier detection that are in existence, it remains to be an all-embracing topic in the research domain. There are still newly proposed methods and essential issues to be addressed. Therefore, this article serves a vital role in keeping researchers abreast with the latest progress in outlier detection techniques. To the best of our knowledge, most surveys conducted so far only address specific areas rather than providing in-depth coverage and insights of up-to-date research studies, as can be seen in Table 1. For example, the review in [25] only focuses on data streams, [27] focuses on high dimensional numeric data, [23], [33] on dynamic networks and the most recent on deep learning [32]. The most comprehensive ones [28], [33], [41], despite containing a lot of insights, they do not review most of the primary state-of-the-art methods, with most published at least five years ago.

In recent years, more contemporary studies have been conducted, especially in the area of deep learning [35], [36] and ensemble techniques [37], [38]. Therefore, more of these recent studies and discoveries needs a review. Our survey presents a comprehensive review of the most prominent state-of-the-art outlier detection methods, including both conventional and emerging challenges. This survey is different from others because it captures and presents a more comprehensive review of state-of-the-art literature, as well as consolidating and complementing existing studies in the outlier detection domain. In addition, we did extensive research to bring forth significant categories of outlier detection approaches and critically discuss and evaluate them. We further discussed commonly adopted evaluation criteria, as well as the tools and available public databases for outlier detection techniques. We believe, this survey will significantly benefit researchers and practitioners as it will give a thorough understanding of various advantages, disadvantages, open challenges, and gaps associated with state-of-the-art outlier detection methods. This will provide them with a better insight into what needs to be focused on in the future. In summary, the novel and significant contributions of the paper are:

- We present the different up-to-date outlier definitions, the different kinds, causes, contemporary detection and handling process, and the latest challenges and

application areas. Unlike other surveys, we add new application areas that need more attention.
- We expand on the categories of outlier detection algorithms with additional distinct methods to previous surveys. We introduce state-of-the-art algorithms, discuss them with highlighting their strengths and weaknesses. We mainly cite and discuss recent studies that were done after most of the significant surveys [26], [33].
- We significantly expand the discussions for each of the distinct categories, in comparison to previous surveys, by presenting the pros, cons, open challenges, and shortfalls of recent methods. We also offer a summary of the performance of some state-of-the-art algorithms, issues solved, drawbacks, and possible solutions.
- We present some of the contemporary open challenges in evaluating outlier detection algorithms. We then introduce standard tools, and some benchmark datasets usually used in outlier detection research. We extend our discussion with a discussion of the OD tools selection and challenges in choosing suitable datasets.
- We identify some challenges and finally recommend some possible research directions for future studies.

The paper is organized as follows: In section 2, we commence our study by providing a comprehensive background on outlier detection. This is done through a detailed explanation about their most significant outlining features and foundations: the definition, characteristics, causes, and application areas. In Section 3, we formally categorize the outlier detection methods (OD) into distinct areas and then discuss these techniques briefly. We include the performances, issues addressed, and drawbacks of these methods with open research questions and challenges for future work. Section 4 contains the discussion of some evaluation constraints in outlier detection, essential tools used for OD, and some analysis of benchmark data sets. In Section 5, we conclude the paper with some open challenges and recommendations for future work.

II. BACKGROUND

In this section, we present commonly used definitions of an outlier, discuss the causes of outliers, new techniques on how to identify and detect outliers, and what to do when an outlier is detected. Finally, we introduce some new application areas of outlier detection and provide additional references for further studies in these application areas.

A. OUTLIER DEFINITION

Since the start of outlier detection research, there have been many definitions of what an outlier is. In 2017, Ayadi *et al.* [14] gave twelve different interpretations of outliers from the perspective of different authors. This demonstrates how complex it is to provide an accurate definition of an outlier. Despite the vagueness and complexity in defining an outlier, it can generally be described as a data point that is significantly dissimilar to other data points or a point that does not imitate the expected typical behavior of the other points [5]. Data points that are contrary to outliers are called inliers. A simple illustrative two-dimensional data set example that depicts an outlier status is shown in Fig. 1.

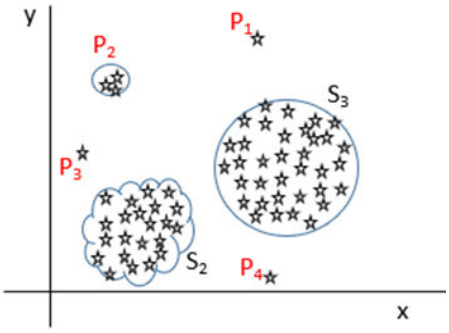


FIGURE 1. A simple example of outliers in the two-dimensional data set.

The data contain two sections, S_1 and S_2. P_1, P_3, P_4, and the other section with very few data points P_2, are far away from the two large clustered regions. Therefore, as per the definition above, they do not conform to the normal behavior of the data and are dissimilar. Thus, they are referred to as outliers.

B. CAUSES OF OUTLIERS, IDENTIFICATION PROCESS, AND HANDLING PROCESS

1) WHAT GIVES RISE TO OUTLIERS AND HOW TO IDENTIFY OUTLIERS

There are quite a lot of different issues that prompt the occurrence of outliers. Some of the most common causes of outliers are as a result of a mechanical fault, changes in system behavior, fraudulent behavior, malicious activity, human error, instrument error, setup error, sampling errors, data-entry error, and environmental changes. For instance, outliers from data errors are usually a result of human error, such as in data collection entry and recording. The next issue with the presence of outliers is how to identify and deal with them.

Many researchers have tried to answer the question of how to detect outliers. The necessary features that need to be considered and the tests that need to be performed to identify the outliers are equally important questions. Even with the growing interest in this research field, there are still on-going studies conducted to find the right answers to these questions. Researchers continue to bring forward novel and innovative ideas to answer them [28], [29]. Over the years, the process of outlier identification carries many names in machine learning and data mining, such as outlier mining, novelty detection, outlier modeling, anomaly detection, etc. In the process of detecting and eliminating outliers, it is important to be observant. Eliminating outliers in correct data might cause the loss of vital hidden information. It is also crucial in the quest of

detecting outliers to know the number of features that need to be considered - a univariate or multivariate case. Also, for a statistical-based approach scenario, whether the selected features can make assumptions of the distribution of values for parametric or non-parametric cases.

Many techniques have been designed to identify outliers, and in Section 3, we will present and discuss further the different recent proposed methods for outlier detection. In this paper, we categorize these outlier identification methods into the following:

- *Statistical-based methods*

The fundamental idea of statistical-based techniques in labeling or identifying outliers depends on the relationship with the distribution model. These methods are usually classified into two main groups - the parametric and non-parametric methods.

- *Distance-based methods*

The underlying principle of the distance-based detection algorithms focuses on the distance computation between observations. A point is viewed as an outlier it is far away from its nearby neighbors.

- *Density-based methods*

The core principle of these methods is that an outlier can be found in the low-density region, whereas inliers are in a dense neighborhood.

- *Clustering-based methods*

The key idea for clustering-based techniques is the application of standard clustering techniques to detect outliers from given data. Outliers are considered as the observations that are not within or nearby any large or dense clusters.

- *Graph-based methods*

Graph-based methods are based on the use of graph techniques to efficiently capture the interdependencies of interconnected entities to identify the outliers.

- *Ensemble-based methods*

Ensemble methods focus on the idea of combining the results from dissimilar models to produce more robust models to detect outliers efficiently. They help to answer the question of whether an outlier should be linear-model based, distance-based, or another kind of model-based.

- *Learning-based methods*

Learning-based methods such as active learning and deep learning, the underlying idea is to learn different models through the application of these learning methods to detect outliers.

2) HOW TO HANDLE OUTLIERS

There is still considerable discussion on what are considered outliers. The most applicable rule of thumb used by many researchers is to flag a data point as an outlier when the data point is three or more standard deviations from the mean [39]. This, however, is a weak supporting idea to discuss such argument further, since it cannot hold for all other scenarios. This is especially true in recent times, when we are faced with large dynamic and unstructured data. Therefore, in modern times, it is imperative to further deliberate on some crucial questions to determine how to handle outliers. For example, whether it is prudent to remove outliers or acknowledge them as part of the data. Outliers in data can sometimes have a negative impact. In machine learning and deep learning outlier detection processes, this will consequently result in longer training process of the data, less accurate models, and eventually degrading results.

With the recent development of new techniques to detect outliers, new approaches have been proposed to deal with outliers. In some cases [42], [43], visual examination of the data is more preferred to get a clear picture of the degree of outlierness of a data point. In another case [44], an approach such as the univariate technique is used to search for data points that contain extreme values on a single variable. While other strategies such as the multivariate technique search for the combinations on the entire variables and then the Minkowski error minimizes prospective outliers during the training phase. There is another great deal of controversy as to what to do when outliers are identified. In many situations, just answering why there are outliers in the data can boost the decision of what can be done with these outliers. In some scenarios, outliers are illegally included [45], while in some other case, they might be part of the data [14]. In cases of high dimensional numeric data computation [27], [46], [47], there are some critical factors like the curse of dimensionality that needs to be considered. Researchers recently tried to use more accurate data that is uncontaminated and ones that are suitable for an outlier detection process [48]–[52], before they start the outlier detection procedure.

Generally, dealing with outliers is dependent on the application domain. For example, in cases where the influence of outliers might cause serious issues such as errors from instrument readings, critical environment safety scenarios, or in real-time situations (fraud detection/intrusion detection). These outliers can be purged, or an alarm is set up. While, in a no cause for alarm scenario, in a case like in a population census survey where few people stand out in some features like height, these outliers can be noted and verified since they are just naturally occurring outliers. There is no need to delete them as in the former case.

In most cases, to answer the question about how to handle outliers, one has to use their intuition, analytic argument through some experiments and also thoughtful deliberation before making decisions. Other noteworthy questions in the outlier detection process, include the significance of considering the context and scenario, and in deliberating the purpose of detecting the outliers. It is essential to know the reason why the outliers are to be identified and what they signify at the end. In the subsequent sections, we will see that different methods or application areas call for various measures on how to deal with outliers.

C. APPLICATION AREAS OF OUTLIER DETECTION

Outlier detection, with its ever-growing interest, has several applications areas in wide-ranging areas. The applications areas where outlier detection is applied are so diverse, it

is impossible to cover thoroughly in just a single survey, because of space limitation. Therefore, in this paper, we list and introduce existing and recent application areas. We will refer our readers to some previous surveys that exhaustively cover many application domains that OD methods are applied in.

Chandola *et al.* [20] provided a broad outline and an in-depth knowledge of outlier detection application domain. Also, the survey [5] also presented an exhaustive list and discussions of applications that adopt outlier detection. Some existing application areas include credit card fraud detection [53], [54], intrusion detection [55], defect detection from behavioral patterns of industrial machines [56], sensor networks [14], finding unusual patterns in time-series data [57], [58], trajectories [19], [59], e-commerce [60], energy consumption [62], data quality and cleaning [16], [45], textual outlier [61], in big data analysis [12], [63], in social media [64], [65] and so on.

Recently, detecting outliers has become essential in these application domains. We consider only a few new application areas of interest for just a short introduction.

1) DATA LOGS AND PROCESS LOGS

Providing outlier detection solutions give companies the edge in gaining concealed insights on their websites, which, if not carried out, will necessitate more effort and additional cost. In processing logs, some automated data mining techniques are needed to search for unusual patterns in the large volume of logs [66]. These logs provide a good source of information for outlier detection monitoring.

2) FRAUD DETECTION AND INTRUSION DETECTION

In fraud detection, if a card is stolen, the purchasing behavior of the card user usually changes; we will notice an abnormal buying pattern. The same is valid for unauthorized access in computer networks, which results in an unusual pattern [55]. Detecting these abnormal (outlier) patterns is essential for security.

3) SECURITY AND SURVEILLANCE

Consider safety and surveillance in the field of cybersecurity. When we take into consideration computer networks, the processes of ensuring safe logging and log administration are very significant as they improve authenticity and security intelligence. Detecting outliers in surveillance videos is a practical and exciting research area [239].

4) FAKE NEWS AND INFORMATION, SOCIAL NETWORKS

In recent times, social media has given a platform for people to spread fake news continually. Sometimes, it is difficult to differentiate between real and fake news. However, from a reliable source, false news reports can be seen as outliers, since they stand out [237]. The spread of fake news has negative influence on people and society at large, so it is also crucial to be identified.

5) HEALTH CARE ANALYSIS AND MEDICAL DIAGNOSIS

In the health care system and medical applications, we usually get unusual patterns or readings from these devices, which generally show a disease condition is diagnosed. The detection and understanding of the abnormal patterns help in the proper diagnosis of the disease and its underlining consequences. It allows doctors to take adequate measures.

6) DATA SOURCES OF TRANSACTIONS

Audit logs for financial transactions contain information about the database operations. The audit logs help in verifying the accuracy, legality, and in reporting the risks. It is essential to monitor the audit logs constantly to identify and report unusual behaviors [67].

7) SENSOR NETWORKS AND DATABASES

Detecting outliers in sensor environments such as in a wireless sensor environment [68], [69], target tracking environment [70], and body sensor networks [71] has helped in ensuring quality network routing and in giving accurate results from sensors. It helps in monitoring the computer network performance, for example, to detect network bottlenecks.

8) DATA QUALITY AND DATA CLEANING

Data from different application areas may contain and generate measurement errors and dirty data. Thus, the process of outlier detection [16], [45], can enhance data quality and cleaning. The method of cleaning and correcting data is essential for training high-quality model and the fast computation and prediction of accurate results.

9) TIME-SERIES MONITORING AND DATA STREAMS

Detecting outliers in time series data [31], [57] and in detecting abnormal patterns in data streaming [10], [25], [72]–[74] is essential. This is because the abnormal pattern will influence the fast computation and estimation of correct results.

10) INTERNET OF THINGS (IOT)

IoT devices are made of a lot of sensors that unceasingly sense environmental parameters. These sensors are successfully fused to obtain information on a specific area or region, depending on the desired task. Before carrying out this task, it is essential to check the quality of data, since the data might be polluted with outliers. It is important to identify or detect these outliers in order not to limit the overall efficiency.

III. OUTLIER DETECTION METHODS

Outlier detection methods have been classified into different techniques such as statistical-based methods, distance-based methods, graphical-based methods, geometric-based methods, depth-based methods, profiling methods, model-based and density-based methods in a wide range of surveys [23], [24]. In this paper, we categorize our outlier detection techniques into six key groups - statistical-based, distance-based, density-based, clustering-based,

ensemble-based, and learning-based techniques. We give a short overview of the different methods and the research progress that has been made in the following categories. Also, we present the advantages, disadvantages, challenges, and some possible future research directions for the different methods. In some approaches, we offer a concise summary in a table format (Tables 2-5) of the various method performances, and issues addressed.

A. DENSITY-BASED APPROACHES

Applying density-based methods to outlier detection is one of the earliest known approaches to outlier detection problems. The core principle of the density-based outlier detection methods is that an outlier can be found in a low-density region, whereas non-outliers (inliers) are assumed to appear in dense neighborhoods. The objects that differ considerably from their nearest neighbors, i.e., those that occur far from their closest neighbors, are flagged and always treated as outliers. They compare the local point's densities with their local neighbor, densities. In density-based outlier detection methods, more complex mechanisms are applied to model the outliers, when compared to distance-based methods. Notwithstanding this, the simplicity and effectiveness of density-based methods have made them widely adopted to detect outliers. Some algorithms designed using this approach have served to be the baseline algorithms [8], [75] for many new algorithms [76]–[78].

Breunig *et al.* [8] proposed the Local Outlier Factor (LOF) method, which is one of the first fundamental loosely related density-based clustering outlier detection methods. The technique makes use of the k-nearest neighbor. In the KNN set of each point, LOF makes use of the local reachability density (lrd) and compares it with those of the neighbors of each participant of that KNN set. The local reachability density (a density estimate that reduces the variability) of an object p is defined as:

$$lrd(p) = 1/\frac{\sum_{o \in kNN(p)} reach - dist_k(p \leftarrow o)}{|kNN(p)|} \tag{1}$$

where

$$reach - dist_k(p \leftarrow o) = \max\{k - dist(o), d(p, o)\} \tag{2}$$

The final local outlier factor score is given as:

$$LOF_k(p) = \frac{1}{|kNN(p)|} \sum_{o \in kNN(p)} \frac{lrd_k(o)}{lrd_k(p)} \tag{3}$$

where $lrd_k(p)$ and $lrd_k(o)$ are the local reachability density of p and o, respectively. The main focus of the approach is that the outlier degree of the observation is defined by its clustering structure in an adjacent neighborhood. The LOF score is at its peak if the *lrd* of the test points is smaller when compared to the nearest neighbors' estimates. Storing the KNN and *lrd* values simultaneously when computing the LOF scores of all data points will incur $O(k)$ additional operation in each point. Therefore, it is prudent to apply a valid index, because in the absence of the useful index, for a data set of size n, it will incur (n^2) time when a sequential search is applied. Because of these shortcomings, Schubert *et al.* [79] found that the LOF density estimate can be simplified, and they proposed a simplifiedLOF to replace the LOF's reachability distance with the KNN distance.

$$dens(p) = \frac{1}{k - dist(p)} \tag{4}$$

where *k-dist(p)* replaces *k-dist(o)* in (3). Even though the SimplifiedLOF showed improved performance, it has a computational complexity similar to LOF.

In a later study, an improvement to LOF [8] and simplifiedLOF [79], is introduced by Tang *et al.* [80], which they called the Connective-based Outlier Factor (COF). The method is closely similar to the LOF with the only difference being the way the density estimation of the records is computed. COF uses a chaining distance as the shortest-path to estimate the local densities of the neighbors while LOF uses the Euclidean distance in selecting the K-nearest neighbors. The drawback to this approach is the indirect assumption made towards the data distribution, which results in incorrect density estimation. The key idea proposed by the authors is based on differentiating "low density" from "isolativity". The isolativity is defined as the degree of an object's connectivity to other objects. The COF value at p with respect to its k-neighborhood is expressed as

$$COF_k(p) = \frac{|N_{k(p)}|ac - dist_{Nk(p)}(p)}{\sum_{o \in Nk(p)} ac - dist_{Nk(o)}(o)} \tag{5}$$

where $ac\text{-}dist_{Nk(p)}$ is the average chaining distance from p to N_k *(p)*. COF adjusts the SimplifiedLOF's density estimate to justify the 'connectedness' via a minimum spanning tree (MST) of the neighborhood. The cost of $O(k^2)$ is incurred when computing the MST of the KNNs. Their method still maintains a similar time complexity as the LOF except in cases where connective data patterns characterize the data sets.

After a couple of techniques, it is still confusing which threshold score can be considered as an outlier in LOF. Kriegel *et al.* [81], then formulated a more robust local density estimate for an outlier detection method called the Local Outlier Probabilities (LoOP) which combines the idea of providing an outlier 'score' with a probabilistic and statistical-oriented approach. It makes use of a density estimation that is based on the distance distribution, and the local outlier score is defined as a probability. LoOP tries to address the issue of LOF outputting an outlier score instead of an outlier probability. The advantage of using the LoOP's probability score is that it may give a better comparison of the outlier records for different datasets. The LoOP showing that a point is an outlier is given as:

$$LoOP_S(O) = \max\left\{0, erf\left(\frac{PLOF_{\lambda,S}(O)}{nPLOF.\sqrt{2}}\right)\right\} \tag{6}$$

where $PLOF_{\lambda,S}(O)$ is the probabilistic local outlier factor of an object with respect to the significance of λ_r a context

set $S(o) \subseteq D$ and *nPLOF* is the aggregated value. Points within the dense region will have a LoOP value close to 0 while those that are closer to 1 will be for density-based outliers. Similar to simplifiedLOF [79], the LoOP normalizes its outlier detection score, which gives it the same complexity of $O(k)$ per point as in [79]. The LoOP, like other previous local outlier algorithms, computes the local density estimation using the neighborhood set. However, computing the density is different. It follows the assumption of a ''half-Gaussian'' distribution and applies the probabilistic set distance (standard deviation).

In LOF [8] and COF [80], these methods fall short of handling the issue of multi-granularity correctly. Papadimitriou *et al.* [82] proposed a technique with the LOcal Correlation Integral called LOCI and its outlier metric - the multi-granularity deviation factor (MDEF), to handle this drawback. Points that deviate at least three standard deviations away from MDEF's neighbor are marked as an outlier. It deals well with the local density variations in the feature space and also detects both distant clusters and secluded outliers. The MDEF of a point p_i at a radius r is mathematically defined as:

$$MDEF(p_i, r, \alpha) = 1 - \frac{n(p_i, \alpha r)}{\hat{n}(p_i, r, \alpha)} \tag{7}$$

where $n(p_i, \alpha r)$ and $\hat{n}$ (p_i, r,α) are the number of αr neighborhood objects and the average of all the objects p in the r-neighborhood of p_i. For the faster computation of the MDEF, if we estimate the value of the numerator and denominator of the fraction on the right-hand side, this gives a better result. All the previous algorithms have shown that it is crucial to choose an appropriate k for excellent detection performance. In the LOCI, a maximization approach is used to address this issue. The method adopts the half Gaussian distribution to estimate the local density; similar to LoOP. However, instead of using the distance, the aggregate of the records in the neighborhood are used. Another point worth noting is that the LoOP has a different way to estimate the local density. It differs from that of LOCI. Instead of comparing the local density's ratio, it examines two different sized neighborhoods. Even though the LOCI showed good performance, however, it has a longer runtime and Papadimitriou *et al.* [82], proposed another method, an approximate version of LOCI called aLOCI. To increase the counting speed of the two neighborhoods, the quad-trees are applied with some constraints.

Another technique when compared with existing methods, LOF [8] and LOCI [82], that performs more efficiently as a result of its pruning ability for data points that are deep in a cluster was proposed by Ren *et al.* [83]. It shows better scalability with an increase in data size. They proposed a method called the Relative Density Factor (RDF) method, and which uses a vertical data model (P-trees) to detect outliers. The RDF is the degree of the measure of outlierness and outliers are points with high RDF values. The RDF of point p is the ratio of the neighborhood density factor of point p divided by its density factor.

$$RDF(p, r) = \frac{DF_{nbr}(P, r)}{DF(P, r)} \tag{8}$$

where *DF(P,r)* is the density factor that is defined as the ratio of the number of neighbors of P and the radius r, while DF_{nbr} *(P,r)* is the neighborhood density factor of the point p.

Jin *et al.* [75] proposed INFLuenced Outlierness (INFLO), which is another technique for local outlier detection similar to that of LOF and uses the symmetric neighborhood relationship to mine outliers. In LOF, for a dataset with closely related clusters of different densities, correctly computing the score of the instances at the cluster borders is not given. INFLO addresses this shortcoming. It solves the problem of inaccurate space representation in the LOF. INFLO uses different descriptions of the neighborhood for the reference set and context set. The INFLO score is computed using both the k-nearest neighbors and the reverse nearest neighbor. To achieve an enhanced estimation of the neighborhood's density distribution, both the nearest neighbors (NN) and reverse nearest neighbors (RNNs) of data points are considered. INFLO is defined as the ''ratio of the average density of objects in $IS_k(p)$ to $p's$ local density'':

$$INFLO_k(p) = \frac{\sum_{o \in IS_k(p)} den(o)}{|IS_k(p)| den(p)} \tag{9}$$

where *den(o)* and *den(p)* are the densities of o and p respectively, and $IS_k(p)$ is the average density of objects to $p's$ local density. The higher the INFLO value, the higher the probability that the object is an outlier. In 2014, still using the density-based approach to tackle local outlier detection problems, Cao *et al.* [84] proposed a novel density-based local outlier detection (UDLO) notion on uncertain data that are characterized by some discrete instances. Here, an exact algorithm is recommended to compute the density of an instance rather than using the naive method of finding all k-neighbors to calculate the outliers, as in the LOF. However, in their approach, they only applied the Euclidean distance metrics. Using other distance computation methods to investigate the possibility of improving the performance of the algorithm can be a future study.

After the introduction of LOF [8], several variations of LOF have been established, such as COF [80], INFLO [75], and LOCI [82]. However, these algorithms are challenged with the distance computations for high dimensional datasets. Keller *et al.* [85] proposed a high contrast subspace method (HiCS) to improve on evaluating and ranking of outliers where outlier scores are closely related. Extending the focus beyond only local outliers to include global outliers, Campello *et al.* [86] proposed a new effective outlier detection measure algorithm called Global-Local Outlier Score from Hierarchies (GLOSH). It is capable of simultaneously detecting both global and local outlier types based on a complete statistical interpretation. Generally, even though the GLOSH result can't perform better in all cases than other techniques, it still has the strength of scaling well for

different tasks. Since the study is based on a specific k-nearest neighbor density estimate, it has some limitations. A future study could be to investigate how other density estimates would improve this work.

Momtaz *et al.* [87], deviate a little from the central focus of most previous algorithms in computing the local outliers. They introduced a novel density-based outlier detection technique that detects the top-n outliers by providing for every object a score called the Dynamic-Window Outlier Factor (DWOF). This algorithm is a modified and improved version of Fan *et al.* [88] - Resolution-based Outlier Factor (ROF) algorithm. ROF overcomes some setbacks, such as low accuracy and its high sensitivity to parameters in data sets.

With the massive flow of high-dimensional data, new research motivations are linked with improving the effectiveness and efficiency of algorithms in detecting outliers in big data. Wu *et al.* [89] proposed an algorithm for the detection of outliers in big data streams. They use a fast and accurate density estimator called RS-Forest and a semi-supervised one class machine-learning algorithm. Bai *et al.* [77], considered a density-based outlier detection in big data and proposed a Distributed LOF Computing (DLC) method, which detects outliers in parallel. The main idea here is twofold. Initially, the preprocessing stage uses the Grid-Based Partitioning (GBP) algorithm and the DLC for the outlier detection stage. However, despite the improved performance, it still does not scale well when compared to Lozano *et al.* [90] - Parallel LOF Algorithm (PLOFA). Improving the scalability of the algorithm can be an interesting research problem for future direction.

Tang and He [78] proposed an outlier detection method using the local KDE. To measure the local outlierness, a Relative Density-Based Outlier Score (RDOS) is used. Here, the local KDE method with an extension of the object's nearest neighbor is used to estimate the density distribution at the object location. They pay more emphasis on the reverse and shared nearest neighbors rather than the k-nearest neighbor of an object for the density distribution estimation. In their method, only the Euclidean distance metric is applied, similar to UDLO in [84]. With a related extension for a future study, there is a need to involve other distance methods to investigate its effect, and to extend their work in real-life applications.

Vázquez *et al.* [91] proposed a novel algorithm to detect outliers based on low-density models of the data called Sparse Data Observers (SDO). SDO reduces the quadratic complexity experienced by most lazy learner OD algorithms. It is an eager learner and severely reduced the computational cost, which in turn performs well when compared to other best-ranked outlier detection algorithms. Ning *et al.* [92] proposed a relative density-based OD method that uses a novel technique to measure the object's neighborhood density. Su *et al.* [93] proposed an efficient density-based scheme based on local OD approach for scattered Data called E2DLOS. They rename the local outlier factor and called theirs Local Deviation Coefficient (LDC) by utilizing the full benefit of the object distribution and the distribution of the neighbors. They then proposed a safe non-outlier object removal method to preprocess the datasets to remove all non-outlier objects. This process is named as rough clustering based on multi-level queries (RCMLQ). This helps in reducing the amount of data that is required to be computed for the local outlier factor. The proposed method is based on LDC and RCMLQ, and from the experiment, it improves on existing local outlier detection methods in both detection accuracy and time efficiency.

We present a summary in Table 2, showing the progress using this technique for some key algorithms mentioned above. In our overview, it is essential to note that, when we say this method outperforms the other, it does not necessarily mean that it is superior to the other in all scenarios and datasets. The analysis and summary presented here are based on the experiment done in these papers, as reported by the authors. While a method might outperform another method, this might be for a set of parameters, the scenario or assumptions used in the experiment. We cannot claim that a method is superior to another in all cases since we did not perform experiments under the same parameter settings and environment. This is true for all the following tables (Table 2-5) in this paper.

1) DENSITY-BASED APPROACHES-ADVANTAGES, DISADVANTAGES, CHALLENGES, AND GAPS

a: ADVANTAGES

In density-based methods, the density estimates used are non-parametric; they do not rely on assumed distributions to fit the data.

Some of the density-based techniques [8], [75], [81], [82] have served as a fundamental baseline for many subsequent algorithms. They have experimentally been shown to work well for modern methods, often outperforming their competitors like some existing statistical and distance-based approaches [39], [94], [95]. Since outliers in these methods are often analyzed through the object's neighborhood density [8], [82], this, in turn, gives it more advantage in identifying crucial outliers missed by most other outlier detection-based methods. These methods facilitate the process of efficiently ruling out outliers nearby some dense neighbors. They require only minimum prior knowledge such as the probability distribution and only a single parameter tuning. They are also known for their ability to compute local outliers efficiently.

b: DISADVANTAGES, CHALLENGES, AND GAPS

Even though some density-based methods are shown to have improved performance, they are more complicated and computationally expensive when compared especially to statistical methods in most cases [96]. They are sensitive to parameter settings such as in determining the size of the neighbors. They need to cautiously take into consideration several factors, which consequently results in expensive computations. For varying density regions, it becomes more

TABLE 2. summary of density-based algorithms.

Methods & Year	Performance	Issues Addressed	Drawbacks
LOF (2000) [8]	With good index: $O(n)$ Without index: $O(n^2)$ Medium dimension: $O(nlogn)$ Extremely high dimension: $O(n^2)$	It addresses the issue of considering outliers as a binary property by assigning an outlier factor to every object to show that it is not a binary property. Provides solution to local density outlier detection problems.	Fails to deal with the multi-granularity problem and is sensitive to the choice of the MinPts. Requires the computation for all objects in the dataset, which is expensive and might miss possible outliers whose local neighborhood density is very close to that of its neighbors.
COF (2002) [80]	$O(n)$ for low-dimension, $O(nlogn)$ for medium dimension and $O(n^2)$ for high-dimensional datasets. It shares similar time complexity as LOF under the same settings.	COF addresses LOF weakness of ruling out outliers closer to non-outlier patterns with low density. It can independently detect outliers of densities of patterns from which they deviate.	It demands more expensive computations than LOF.
LOCI (2003) [82]	It shows a linear scale for the dataset size as well as the dimension. However, the computational complexity for computing the neighbors is $O(n^3)$ and $O(n^2)$ in terms of memory.	Addresses some drawbacks such as: the sensitive to the choice of parameters, dealing with both local density and multi-granularity. In addition, dealing with large sampling neighborhoods.	It incurs an extra computational cost to compute the standard deviation.
RDF (2004) [83]	For large data size, it is more scalable and takes less time when compared to LOF and aLOCI.	Based on the RDF, it prunes the data points that are deep in clusters and detect outliers that are only within the remaining small subset of the data.	For a small size data set, the running time is slower when compared to other existing methods [8] [99].
INFLO (2006) [75]	For smaller data size and dimension (d), it has less running time, but when $d>=12$, the runtime increases and makes the computational time expensive.	To calculate its density ratio, it uses only the data points in its k-influence space, which in turns makes the outliers more meaningful. Therefore, it shows that it can identify more meaningful outliers than LOF.	It is designed only for local outliers, based on a symmetric neighborhood relationship.
LoOP (2009) [81]	Its performance is relatively enhanced and much more stable for larger range values. It is also more robust against its competitors in terms of the choice of k.	LoOP values are steadier over the complete dataset and multiple datasets. It offers each data object an outlier probability score that is simple to interpret and can be compared over single and dissimilar even datasets.	LoOP techniques do not focus on the efficiency of the method instead it focuses on the recall and precision of the method.
DWOF (2013) [87]	It shows improved performance in terms of both the accuracy and sensitivity to the number of parameters than LOF.	It overcomes the limitations of low accuracy and high sensitivity to parameter k. It also further solves the issue of duplicates in the window.	The computational cost of the method was not evaluated; only the detection accuracy was focused on.
GBP+ DLC (2016) [77]	It shows improved performance when compared with PLOFA [90] in terms of its processing time and data delivery quality.	It addresses the challenge faced by centralized environments, which with an increasing amount of data, the algorithm processing efficiency becomes limited. They solve this issue by detecting the outliers in a distributed manner.	With the growth of data size, the runtime becomes longer and therefore, not very scalable.
RDOS (2017) [78]	For large scale data sets, the algorithm's area under the curve shows better performance than LOF in detecting local outliers.	It uses an effective relative density calculation based on kernel density estimation to measure the outlierness. For a more robust computation of the outliers, instead of considering only the k-nearest neighbor as in the case of other techniques. They applied k-nearest, reverse nearest, and shared nearest neighbors.	It lacks the use of other distance methods as it uses only a single distance metrics - the Euclidean distance metrics.

complicating and results in poor performance. Density-based methods, due to their inherent complexity and the lack of update of their outlierness measures, some of these algorithms, such as INFLO and MDEF, cannot resourcefully handle data streams. In addition, they can be a poor choice for outlier detection in data stream scenarios. It is also challenging for high dimensional data when the outlier scores are closely related to each other.

To discuss further, in Table 2, we present a summary of randomly handpicked (because of the space limitation) well-known density-based outlier detection algorithms. We included the performance, issues addressed, and drawbacks of standard algorithms and show the progress of how these algorithms have evolved. In one of the most popularly known density-based methods, LOF [8], it is crucial to note that in an outlier detection process where the local outliers are not significant, the algorithm can create a lot of false alarms. Generally, since density-based methods are non-parametric, for high-dimensional data spaces, the sample size is considered too small [27]. Additional re-sampling, to draw new samples, can be adapted to enhance the performance. We also note that, since most density-based methods rely on nearest neighbor computations, this makes the choice of k very significant for the evaluation of these algorithms. Usually, finding the nearest neighbor in nearest-neighbor outlier detection algorithms, the computational cost is about $O(n^2)$. A rare case is that in LOCI, where the radius r is extended and thus summing its complexity to $O(n^3)$. This makes it very slow for more massive datasets. An improved version is the aLOCI, which shows a faster runtime depending on the amount of the quad-trees that are utilized. Goldstein *et al.* [97], compared COF and LOF, and it was found that the LOF spherical density estimation was a poor choice for efficiently detecting outliers. COF estimated its local density by connecting the regular records with each other to solve the above drawback. INFLO shows improved outlier scores when clusters with different densities are not far from each other. Table 2 gives the remaining summary of critical points for the different algorithms.

B. STATISTICAL-BASED APPROACHES

Detecting outliers using statistical techniques can be done using supervised, semi-supervised, and unsupervised styles. In statistical-based OD methods, the data points are sometimes modeled using a stochastic distribution, and some data points can be labeled as outliers depending on the relationship with the distribution model. Outliers and inliers are declared depending on the data distribution model. Statistical-based methods are usually classified into two main groups - the parametric and non-parametric methods. The major difference between the two methods is that the former has an assumption of the underlying distribution model in given data, and from the known data, it estimates the parameters of the distribution model. The latter method does not have any assumption of prior knowledge of the distribution model [98]. In this paper, we classify some of the current research that has been done using a statistical approach to detect outliers into three categories - parametric methods, non-parametric methods, and other kinds of statistical techniques.

1) PARAMETRIC METHODS

For this type of method that has an assumption of the underlying distribution model, two well-known methods adopted for outlier detection are the Gaussian Mixture model and Regression model.

a: GAUSSIAN MIXTURE MODEL METHODS

The Gaussian Model is one of the most prevalent statistical approaches used to detect outliers. In this model, the training phase uses the Maximum Likelihood Estimates (MLE) method [100] to perform the mean and variance estimates of the Gaussian distribution. In the test stage, some statistical discordancy tests (box-plot, mean-variance test) are applied.

Yang *et al.* [101], introduced an unsupervised outlier detection method with the globally optimal Exemplar-Based GMM (Gaussian Mixture Model). In their technique, they first realized the global optimal expectation maximization (EM) algorithm to fit the GMM to a given data set. The outlier factor for every data point is considered as the sum of the weighted mixture proportions with the weight signifying the relationship with other data points. The outlier factor F_k at x_k is mathematically defined as:

$$F_k = z_k(t_h) = \sum_{j=1}^{n} s_{kj}\pi_j(t_h) \tag{10}$$

where $s_{kj}\pi_j(t)$ shows the depth of the point x_k 's influence on another point x_j. s_{kj}, referring to the connection strength, t_h the final iteration and π_j the measure of the significance of point j. The data point x_k, is more likely to be flagged as an outlier if F_k, is smaller. This technique is in contrast to other existing methods [8], [80], [82], which focus solely on local properties rather than global properties. Yang *et al.* [101] technique can be applied to solve the problem of the clustering-based technique's inability to detect outliers in the presence of noisy data, by fitting the GMM at every data point in a given dataset. We note that, notwithstanding the method's capacity to identify unusual shapes faster, it still has a high complexity (with $O(n^3)$ for single iteration and $O(Nn^3)$ for N iterations). An algorithm that can reduce such a computational complexity can serve to be more scalable for future study.

In 2015, for a more robust approach to outlier detection, the use of GMM with locality preserving projections was proposed by Tang *et al.* [102]. They combined the use of GMM and subspace learning for robust outlier detection in energy disaggregation. In their approach, the locality preserving projection (LPP) of subspace learning is used to preserve the neighborhood structure efficiently and then reveal the intrinsic manifold structure of the data. The outliers are kept far away from the normal sample, which happens to be reversed when compared to Saha *et al.* [103] principal

component analysis (PCA) method. This study addresses the research gap of the previous methods, LOF [8] and Tang *et al.* [80], which failed to detect outliers in multiple state processes and multi-gaussian states. From the experimental evaluation, even though the proposed method showed improved performance (true-positive 93.8% to 97% and a decrease of false-positive from 35.48% to 25.8%). However, missing in the literature is the computational complexity when compared to other techniques.

b: REGRESSION METHODS

Detecting outliers using regression models is one of the most straightforward approaches to outlier detection problems. The model chosen by the user can either be linear or non-linear depending on the problem that needs to be solved. Usually, when adopting this technique, the first stage, which is the training stage, involves constructing a regression model that fits the data. The test stage then tests the regression model by evaluating every data instance against the model. An outlier here is labeled when a data point with a remarkable deviation occurs between the actual value and the anticipated value produced by the regression model. Over the years, some standard approaches for outlier detection using the regression techniques include thresholding using Mahalanobis distance, robust least squares with bi-square weights, mixture models and then an alternate vibrational Bayesian approach to regression [26]. These techniques use regression models to detect outliers, and in contrast, a different method was proposed by Satman [104] to detect outliers in linear regression. The algorithm is centered on a non-interactive covariance matrix and concentration steps applied in the least trimmed square estimation. The algorithm has the advantage of detecting multiple outliers in a short time, which makes the computational time to be cost-effective. However, for a better result of this model, a future study can be to minimize the bias and the variance of the intercept estimator because regression models are sometimes characterized by minute preferences.

Park *et al.* [105], proposed another regression-based outlier detection technique, but this time, it is centered on detecting outliers in sensor measurements. The proposed technique makes use of a weighted summation approach for building a synthesized independent variable from the observed values. Since the method was only tested for a single environment, we believe proposing techniques that will attain precise model estimation for different sensor settings and situation will be an interesting future study. Recently, in 2017, Dalatu *et al.* [106] did a comparative study on linear and non-linear regression models for outlier detection by analyzing the receiver operating characteristic (ROC) curves in terms of their misclassification rate and accuracy. The study gives researchers insight into the predictive results of the two kinds of regression models in outlier detection. The non-linear models (93% accuracy) tend to fit more than the linear models (68% accuracy) for outlier detection, which gives researchers better reasons why adopting a non-linear model can be more effective in a more general situation.

2) NON-PARAMETRIC METHODS

Kernel Density Estimation Methods: Kernel Density Estimation (KDE) is a common non-parametric approach for detecting outliers [107]. An unsupervised approach to outlier detection using kernel functions was presented in [108] by Latecki *et al.* The outlier detection process is performed by comparing each point's local density to that of the neighbor's local density. The experimental evaluation of the proposed techniques when compared to some popular density-based methods [8], [82] results in better detection performance in most cases. However, the method still lacks applicability in very large and high dimensional real-life databases. This can be an extension of the current study for the future. Later, Gao *et al.* [109] proposed a better approach to address some of the previous shortcomings. The method shows improved performance, and good scalability for broad data sets using kernel-based techniques with less computational time when compared to LOF and Latecki *et al.* [108] proposed methods. To address the issue of inaccurate outlier detection in complex and large data sets, they adopted the variable kernel density estimation to tackle this problem. Another issue to address is related to the LOF, which is the dependability of the parameter k – which measures the weight of the local neighborhood. To salvage this issue, they adopted a weighted neighborhood density estimate. Overall, the method shows improved performance and good scalability for large data sets with less computational time. Kumar and Verma [110] use KDE to estimate the sensor data distribution to detect malicious nodes.

In another study, Boedihardjo *et al.* [111] adopt the KDE based approach in a data stream environment despite the challenges of directly applying the KDE methods in a data stream environment. The KDE methods in outlier detection approaches show improved performance in some aspects. However, they are known for their extensive computational cost. Uddin *et al.* [112] then use KDE for outlier detection in different application area - power grid. The authors in [111] proposed an approximation approach of the adaptive kernel density estimator (AKDE) for robust and accurate estimates of the probability density function (PDF). Although it shows that the technique produces a better estimation quality than the original KDE - with a ($O(n^2)$) computational cost. However, it still shows a better performance in most areas when compared to the original KDE. The authors were able to propose a technique that met the stringent constraints in this kind of environment, and we believe further studies for multivariate streams can be done. Zheng *et al.* [10] in another study, use KDE on distributed streams in a multimedia network for detecting outliers. Smrithy *et al.* [113] proposed a non-parametric online outlier detection algorithm to detect outliers in big data streams. An adaptive kernel density-based technique using the Gaussian kernel was also studied by Zhang *et al.* [114] for detecting anomalies in non-linear systems. Qin *et al.* [115] proposed a novel local outlier semantics that makes excellent use of KDE to detect local outliers from data streams effectively. Their work addresses

the shortcoming of existing works that are not well furnished to tackle current high-velocity data streams owing to high intricacy and their unpredictability to data updates. They designed KELOS, an approach to unceasingly identify the top-N KDE-based local outliers over streaming data.

To conclude, one big setback of most KDE methods is that they usually suffer from a high computational cost and curse of dimensionality, which makes them very unreliable in practice. Despite KDE's better performance when compared to other non-parametric OD approaches, there is a relatively low number of reports that adopt KDE based phenomenon to approach this problem.

3) OTHER STATISTICAL METHODS

Many statistical approaches have been proposed, but among the more straightforward statistical methods to identify outliers are the histogram [116] and other statistical tests [40] such as the Boxplot, Trimmed mean, Extreme Studentized Deviate and the Dixon-type test [40]. The Trimmed mean among the others is more comparatively resistance to outliers, while to identify single outliers, the Extreme Studentized Deviate test is the right choice. The Dixon-type test has the advantage of performing well with a small sample size because there is no need to assume the normalcy of the data. Barnett *et al.* [39] discuss several tests for the optimization of different distributions model to effectively detect outliers. Optimization could depend on the actual parameters of conforming distributions, that is, the expected space for outliers and the number of outliers. Rousseeuw and Hubert [117] also gave a broader discussion of statistical techniques for outlier detection. Using a histogram-based approach, Goldstein and Dengel [116] proposed a Histogram-Based Outlier (HBOS) detection algorithm that uses static and dynamic bin width histograms to model univariate feature densities. These histograms are then used to calculate the outlier score for each of the data instances. Though the algorithm showed improved performance in some performance metrics like the computational speed when compared to some other popular OD methods such as LOF [8], COF [80], and INFLO [75]. However, it falls short in local outlier detection problems because the algorithm cannot model local outliers using the proposed density estimation.

Hido *et al.* [95], proposed a new statistical approach for inlier-based outlier detection problems by using the directed density ratio estimation. The main idea is to utilize the ratio of the training and test data densities as outlier scores. The method of unconstrained least-squares importance fitting (uLSIF) was applied because it is more suitable with natural cross-validation measures that allow it to accurately optimize the tuning parameter's value; such as the kernel width and regularization parameter. The proposed technique, when compared to the non-parametric KDE, is more advantageous because it has provision to escape the hard density estimation computation. The method also showed an improved performance in accuracy, even though not in all cases, they showed a better performance than the other methods. Nevertheless, it demonstrates that the approach is more efficient in a broader perspective. Improving the accuracy of the density ratio estimation can be an important future work to consider this approach.

Du *et al.* [118], proposed another robust technique with statistical parameters to solve the problem of local outlier detection called the Robust Local Outlier Detection (RLOD). This study was motivated by the fact that most OD methods focus on identifying global outliers, and most of these methods [119], [120] are very sensitive to parameter changes. The whole idea of the framework is in three stages. In the first stage, the authors propose a method to initially find density peaks in the dataset using the 3σ standard. In the second stage, in the dataset, each remaining data object is then allocated to its identical cluster to be labeled as to its nearest neighbor with a higher density. In the final stage, they use Chebyshev's inequality and then the density peak reachability to recognize the local outliers in each group. The method supports both the detection of local and global outliers as in Campello *et al.* [86] technique, and they experimentally proved that the method outperforms other methods [8], [26] in terms of the running time and detection rate. The authors recommend further experiments on how to improve efficiency through the use of a robust method for distributed and parallel computing.

Other studies have been done using statistical methods for computing outliers. In Table 3, we present a summary showing the progress using this technique for some key algorithms mentioned above.

4) STATISTICAL-BASED APPROACHES-ADVANTAGES, DISADVANTAGES, CHALLENGES, AND GAPS

Advantages

i. They are mathematically acceptable and have a fast evaluation process once the models are built. This is because most models are made in a compacted form, and they showed improved performance given the probabilistic model.
ii. The models generally fit quantitative real-valued data sets or some quantitative ordinal data distributions. The ordinal data can be changed to an appropriate value for processing, which results in improved processing time for complex data.
iii. They are easier to implement even though limited to specific problems.

Disadvantages, Challenges, And Gaps:

i. Because of their dependency and the assumptions of a distribution model in parametric models, the quality of the results produced is mostly unreliable for practical situations and applications due to the lack of preceding knowledge regarding the underlying distribution.
ii. Since most models apply to univariate feature space, they are characteristically not applicable in a multi-dimensional scenario. They incur high computational costs when dealing with multivariate data and this,

in turn, makes most of the statistical non-parametric models a poor choice for real-time applications.

iii. In the histogram technique, one fundamental shortcoming for multivariate data is the ineptitude of capturing the interactions between different attributes. This is because they cannot simultaneously analyze multiple features. In general, some prevailing statistical methods are not applicable to handle very high dimensional data. There is a need to design statistical techniques to support high dimensional data that are capable of simultaneously analyzing multiple features.

iv. When faced with problems of increased dimensionality, statistical techniques adopt different methods. This results in increase in the processing time and misrepresentation of the distribution of the data.

Discussing further on a more global view, statistical methods comes with lots of advantages and drawbacks. In outlier detection problems, the importance of outlier-free data is significant for building reliable systems. This is because outliers can have a drastic effect on the system efficiency, so it's prudent to identify and remove those that affect the system's accuracy. Most of the drawbacks from statistical methods are centered around the outlier detection accuracy, lack of efficient techniques for very high data sets, the curse of dimensionality, and computational cost.

Statistical-based methods can be effective in the outlier detection process when the correct distribution model is captured. In some real-life situations, for instance, in sensor stream distribution, there is no prior knowledge available to be learned. In such a scenario, when the data does not follow the predetermined distribution, it may become impractical. Therefore, non-parametric methods are mostly appealing since they do not depend on the assumption of the distribution characteristics. This is also true for big data streams, where the data distribution cannot be assumed. For evenly dispersed outliers in a dataset, using statistical techniques becomes complicating. Therefore, parametric methods are not applicable for big data streams, but for non-parametric methods, they are. In addition, defining the threshold of a standard distribution to differentiate the outliers has a higher probability of inaccurate labeling.

For parametric cases, using the Gaussian mixture models, a worth noting point is the daunting task in adopting Gaussian techniques for computing outliers in both a high dimensional data subspace and data streams. Yang *et al.* [101] method, for instance, has high complexity. An algorithm that can reduce such a computational complexity can serve to be more scalable. Regression techniques also are not suitable to support high dimensional subspace data. For a more efficient and robust solution in finding and discovering outliers, it's more appropriate to apply robust regressions rather than ordinary regressions because outliers can impact the latter. For the non-parametric case, KDE performs better in most cases, despite its sensitivity to outliers and the complexity in determining a good estimate of the nominal data density in polluted data sets. In multivariate data, they scale well and are computationally inexpensive. The histogram models work well for univariate data but not suitable for multivariate data. This is because it cannot capture the relations between the different attributes.

Some statistical techniques are not well adapted in recent times because of the kinds of data and application areas. However, they are considered to be great practical approaches to outlier detection problems. Tang *et al.* method [102], when compared to the PCA method [103], gives a robust improvement for outlier and noise detection problems. In HBOS [116], their approach shows a good computational speed even more than some clustering-based algorithms and other types of algorithms (LOCI, LOF, INFLO) and thus makes it a suitable for large scale near real-time applications. While Hido *et al.* [95] method is more scalable for massive data sets and Du *et al.* [118] method has a more robust analysis.

C. DISTANCE-BASED APPROACHES

Distance-based methods detect outliers by computing the distances between points. A data point that is at a far distance from its nearest neighbor is regarded as an outlier. The most commonly used distance-based outlier detection definition is centered on the concept of the local neighborhood, k-nearest neighbor (KNN) [121], and the traditional distance threshold. One of the earliest studies of computing distance-based outliers by Knorr and Ng [122] defined distance-based outliers as:

Definition: In a dataset T, an object O is a DB (p, D)-outlier if minute fraction p of the objects in the dataset lies beyond the distance D from O.

Other well-known definitions of distance-based outliers given the distance measure of feature space, define outliers as:

i. Points with less than p different samples within the distance d [99].

ii. The top n examples whose distance to thekth nearest neighbor are the greatest [123].

iii. The top n examples whose average distance to the k nearest neighbors are the greatest [7].

The abbreviation *DB(p,D)* is the Distance-Based outlier detected using the parameters p and D.

DB outlier detection methods are moderate non-parametric approaches that scale well for large data sets with a medium to high dimensionality. In comparison with statistical techniques, they tend to have a more robust foundation and are more flexible and computationally efficient. In our subsequent section, we classify the distance-based methods into the following groups - distance-based computation method using k-nearest neighbor computation, pruning techniques, and data stream related works. Some of the most commonly used distance-based approaches to detect outliers are as follows:

1) K-NEAREST NEIGHBOR METHODS

Using these methods for computing outliers have been one of the most popular ways adopted by many researchers to detect outliers. It is not the same as the k-nearest neighbor classification. These methods are mostly used for detecting global outliers. Initially, a search for the k-nearest neighbor of every record, and then these neighbors are used to compute the outlier score. They mainly examine the nature of a given object neighborhood information to determine whether they are close to their neighbors or have a low density or not. The key concept is to exploit the neighborhood information to detect the outliers.

Knorr and Ng [122] and Ramaswamy *et al.* [123] were among the first to propose techniques for detecting outliers in large data sets that shows significant progress in the already state-of-the-art existing studies. Knorr and Ng [122], proposed a non-parametric approach, which is in contrast to some previous statistical techniques [101], [104]. The users lack knowledge about the underlying distribution. The indexed-based and nested-loop based algorithms were the two algorithms proposed with the computational complexity of $O(kN^2)$; with k as the dimensionality and N as the number of datasets. Later Ramaswamy *et al.* [123], proposed a cell-based algorithm that is linear with respect to N and exponential with respect to K to optimize the previous algorithm [122]. It has a computational complexity lower than the two previous methods. Ramaswamy *et al.* [123] tried to improve on several of the shortcomings of [122] such as specifying the distance, the ranking method adopted and in minimizing the computational cost. To address the problem of determining the distance, they defined their approach as one that does not require the users to specify the distance parameter but adopts the kth nearest neighbor. In the expanded version of [122], to find the nearest neighbor of each candidate spatial indexing structures, the KD-tree, X-tree, and R-tree are used [99]. This is done by querying the index structure for the closest k points in each example and finally, in line with the outlier definition, the top n candidate is selected. One main concern of this method is that the index structures breakdown with an increase in the dimensionality.

Angiulli *et al.* [7] differ a bit from the traditional approach of targeting the development of techniques to detect outliers in an input dataset, to that which can learn a model and predict outliers in an incoming dataset. They designed a distance-based algorithm that detects top outliers from a given unlabeled dataset and predicts if an undetected data point is an outlier. The outlier detection process involves detecting the top n outliers in a given dataset, which means the n objects of the dataset with the highest weight. This is done by determining whether an incoming object's weight in the dataset is greater than or equal to the nth highest weight. This process results in a $O(n^2)$ complexity.

Ghoting *et al.* [124] proposed an algorithm called the Recursive Binning and Re-Projection (RBRP) to enhance the computational speed for high-dimensional datasets and improve on the drawbacks of previous methods [122], [123]. The key difference from the earlier algorithms is that it supports the fast merging of a point's approximate nearest neighbors. In terms of its efficiency, only the points' approximate nearest neighbors are of value. It scales linearly as a function of the number of dimensions and log-linear for the number of data points. One key difference from other methods is, instead of using the nearest neighbors, the approximate nearest neighbor is used, which makes the computation faster.

In 2009, instead of following the trend in outlier detection for global outliers using distance-based computation techniques, the authors decided to divert to local outlier detection. Zhang *et al.* [76] proposed a local distance-based outlier detection method called the Local Distance-based Outlier Factor (LDOF). Their study shows improved performance over the range of neighbor size when compared to LOF [8]. The demand for a pairwise distance computation is $(O(k^2))$, similar to COF [80]. It is comparable to that of the k-nearest neighbor outlier detection techniques in performance; however, it is less sensitive to parameter values. Liu *et al.* [125], in a later study, extended the traditional LOF to uncertain data.

Huang *et al.* [126] proposed a method called Rank-Based Detection Algorithm (RBDA) to rank the neighbors. It provides a feasible solution and ensures that the nature of high-dimensional data becomes meaningful. For illustration, in [17], the fundamental assumption will be that objects will become close to each other or share similar neighbors when they are produced from the same mechanism. The RBDA uses the ranks of individual objects that are close as the degree of proximity of the object. It does not take into consideration the objects distance information with respect to their neighbors. Bhattacharya *et al.* [127] propose a method that further uses both the ranks of the nearest neighbors and the reverse nearest neighbors. This ensures each candidate object outlier score is effectively measured.

In another study, Dang *et al.* [121] applied k-nearest neighbor to detect outliers in daily collected large-scale traffic data in some advanced cities. Outliers are detected in data points by exploiting the relationship among neighborhoods. An outlier here is a data point that is farther from its neighbors. Notwithstanding the good results shown concerning the success detection rate of 95.5% and 96.2% respectively, which outperforms those of statistical approaches such as KDE (95%) and GMM (80.9%). However, a shortcoming of their work is they only considered a single distance-based metric, so a future study with more complicated variations like that of different weights on multiple distances, can improve the outlier detection rate. In another study, to improve on the search effectiveness of the KNN neighbors, Wang et al. [128] applied a minimum spanning tree.

Radovanovi'c *et al.* [129] presented a reverse nearest neighbor approach to tackle one of the biggest challenges in computing outliers in high-dimensional data sets, that is, ''the curse of dimensionality.'' They showed that their approach could be effectively applied in both low and high-dimensional settings. When compared with the original

KNN method [123], it showed an improved performance in the detection rate. Their primary focus is centered on the influence of high dimensionality and the hubness phenomenon. An antihub technique was then proposed, which optimized the perception between scores. Huang *et al.* [130] implemented the concept of natural neighbor to acquire the information of the neighborhood. Ha *et al.* [131] proposed a heuristic approach to determine a suitable value for k by employing iterative random sampling. To this end, most recently, Tang *et al.* [78] proposed a method to determine the outlier scores in local KDE. They examine different types of neighborhoods, including the reverse nearest neighbor, shared nearest neighbors and the k nearest neighbor. The neighbor-based detection methods are independent of the data distribution model and can be easily understood and interpreted. However, they are sensitive to parameter settings and sometimes deficient in performance.

2) PRUNING METHODS

Bay *et al.* [132], presented an algorithm based on a nested loop that uses the randomization and pruning rule. By modifying the nested loop algorithm, which is recognized for its quadratic performance $O(N^2)$, they were able to obtain a near linear time on most of the data sets that previously showed a quadratic performance in the previous method [122]. However, the algorithm makes a lot of assumptions which will consequently lead to poor performance. Angiulli *et al.* [133], since most previous research [99], [122], [123] were unable to simultaneously meet the demand of both the CPU cost and in minimizing the I/O cost, the authors presented a novel algorithm called Detecting OutLiers PusHing data into an Index (DOLPHIN) to address these challenges. In the proposed algorithm, only two sequential scans of the data set are performed, while that of [132] implements a block-nested loop analysis of the disk pages, which results in a quadratic input and output cost. Ren *et al.* [134], presented an improved version of Ramaswamy *et al.* technique [123], a vertical distance-based outlier detection method to detect outliers in large data sets by also applying the pruning method and a ''by-neighbor'' labeling technique. In their study, as an alternative to the outdated horizontal structures, the vertical structure is adopted to facilitate the efficient detection of outliers. The technique is implemented in two phases (with and without pruning) with P-Trees for the outlier detection. According to the authors, a future study can be to discover the use of P-Trees in other OD methods, such as the density-based approach. In another work, Vu *et al.* [135] introduced the MultI-Rule Outlier (MIRO) that adopts a similar technique as in [134] by using the pruning technique to speed up the process of detecting outliers.

3) IN DATA STREAMS

Lately, most incoming data are in the form of continuous flow, and storing these data can be impractical because they need to be computed fast. In data streams, for distance-based approaches, researchers continue to face significant challenges such as the notion of time, multi-dimensionality, concept drift, and uncertainty problems [72]. Researchers have seen these as interesting challenges, and they have focused on designing algorithms to detect outliers in the data stream environment. The data stream is considered to be a large volume of unlimited incoming sequence data. Since the mining of these kinds of data is highly dependent on time intervals, usually the computation is done in windows. The two well-known data stream window models are the landmark and sliding window [136]. In the former, a time point in the data stream is identified, and the points within both the last time point and the current time are then analyzed. While in the latter, the window is marked by the two sliding endpoints.

Angiulli *et al.* [136] propose a novel idea for the one-time query of outliers in data streams that is different from the continuous queries approach presented by authors in [137], [138]. They proposed three kinds of Stream Outlier Miner (STORM) algorithms to detect outliers in data streams using the distance-based method. The first one is based on computing the exact outlier query and the other two focus on retrieving the approximate results of the queries. The exact algorithm (Exact-Storm) makes use of the stream manager (which collects the incoming streams) and a suitable data structure (that is used by the query manager to answer outlier queries). One shortcoming of this algorithm is the cost of storing all the window objects. It is also not suitable in cases of colossal memory since it cannot fit into the memory. To tackle this issue, the approximate algorithm (Approx-Storm) is applied to improve the Exact-Storm. This is done by adjusting two approximations, that is, by reducing the number of data points stored in each window and by decreasing the space for every data point neighbor's storage. The final algorithm (Approx-fixed-memory) aims to minimize memory usage by keeping only a controlled fraction of the safe inliers.

Yang *et al.* [139], proposed some methods (Abstract-C, Abstract-M, Exact-N, and Extra-N) to deal with the incremental detection of neighbor-based patterns in the sliding window scenarios over data streams. The old static approach of pattern detection is costly and results in high complexity. Therefore, in this technique, the authors address the issue of handling sliding windows, which was not supported in earlier incremental neighbor-based pattern detection algorithms such as the incremental DBSCAN [26]. From their experimental studies, it shows less CPU usage, and it maintained a linear memory usage for the number of objects in the window. Among these algorithms, Abstract-C is the only related algorithm using distance-based while the other two are more linked with density-based cluster methods. Table 3 gives further details and summaries of these methods.

Kontaki *et al.* [140], proposed algorithms that tackle some issues in event detection in data streams [141] and in sliding window scenarios over data stream [139], which are both characterized by continuous outlier detection. In Angiulli *et al.* technique [141], two of the algorithms use the sliding window in parallel with the step function

TABLE 3. summary of statistical-based algorithms.

Methods	Performance	Issue/s addressed	Drawbacks
Gaussian Mixture Models			
Yang et al. [101]	$O(n^3)$ for single iteration and $O(Nn^3)$ for N iterations.	This technique is in contrast with other existing methods [8] [80], which focus solely on local properties rather than global properties [120]. It addresses both properties.	High complexity.
Tang et al. [102]	Shows improved detection accuracy for true-positive and a decrease of false-positive.	Address the issue of algorithms that fail to detect outliers in multiple state processes and multi-gaussian states.	It has no performance measure of the computational complexity.
Regression Models			
Satman [104]	The computational time is cost-effective.	Detects multiple outliers in a short time.	The bias and the variance of the intercept estimator are not well minimized.
Dalatu et al. [106]	Shows higher accuracy for the non-linear models and an average accuracy for the linear models.	It compares the performance of linear and non-linear models for outlier detection.	Not very detailed, in given a definite conclusion for the two models.
Kernel Density Methods			
Latecki et al. [108]	It shows better detection performance than LOF [8] and LOCI [82] in most cases.	Local outlier detection using a density-based approach.	Not applicable for very large and high dimensional real-life data.
Gao et al. [109]	Shows improved performance and good scalability for large data sets with less computational time.	The inaccurate detection of outliers in the complex and large dataset and the dependability of the parameter k – which measures the weight of the local neighborhood.	The complexity of the method.
Boedihardjo et al. [111]	It has a computational cost of $(O(n^2))$ which exceeds that of KDE$(O(n))$.	Adopt KDE based approach in a data stream environment.	Extensive computational demand.
Other Statistical Methods			
Hido et al. [95]	Shows improved performance in wider perspective than LOF in terms of accuracy.	A new statistical approach for inlier-based outlier detection.	The accuracy of the density ratio estimation.
Du et al. [118]	Experimentally proven to outperform LOF [8] and DBSCAN [26] in terms of the running time and detection rate.	Their method supports both the detection of local and global outliers.	The efficiency of the method.

in the process of detecting the outliers. The main objective in [140] is to minimize the storage consumption, improve the algorithm efficiency, and to make it more flexible. The authors designed three algorithms to support their aim, and they include Continuous Outlier Detection (COD), Advance Continuous Outlier Detection (ACOD), and Micro-Cluster-Based Continuous Outlier Detection (MCOD). The first algorithm, COD, has double versions that support a fixed radius and multiple k values, while ACOD supports multiple k and R values. The final algorithm, MCOD, minimizes the range queries and thereby reduces the amount of distance computation that needs to be done. K is the parameter for the number of neighbors, and R is the distance parameter for the outlier detection. The key dissimilarity between STORM and COD is the decline in the number of examined objects in each slide, while for Abstract-C and COD, they are the speed and memory consumption. That is, it is much faster and requires less space. Another algorithm that is designed specifically for a high-volume of data streams was proposed by Cao *et al.* [142] called ThreshLEAP. It is a technique that tries to mitigate the expensive range queries. This is achieved by not storing data points in the same window like those in the same index structure. Leveraging modern distributed multi-core clusters to improve the scalability of

detecting the outliers can be an exciting direction for future studies.

In Table 4, added to the survey done by Tamboli *et al.* [25] in comparing some distance-based outlier detection algorithms using the Massive Online Analysis tool [143], we added other methods that were not included in their work. In addition, Tran *et al.* [73], performed an evaluation study with detailed experiments of outlier detection methods in the data stream. Among all the algorithms presented, they conclude that MCOD in most settings has the best performance.

4) DISTANCE-BASED APPROACHES-ADVANTAGES, DISADVANTAGES, CHALLENGES, AND GAPS

Advantages:

i. They are straightforward and easy to comprehend as they mostly do not rely on an assumed distribution to fit the data.
ii. In terms of scalability, they scale better in a multi-dimensional space as they have a robust theoretical foundation, and they are computationally efficient when compared to statistical methods.

Disadvantages, Challenges, And Gaps:

i. They share some similar drawbacks as statistical and density-based approaches in terms of high dimensional space, as their performance declines due to the curse of dimensionality. The objects in the data often have discrete attributes, which makes it challenging to define distances between such objects.
ii. The search techniques such as the neighborhood and KNN search in high-dimensional space when using a distance-based approach is an expensive task. In large data sets, the scalability is also not cost effective.
iii. Most of the existing distance-based methods that cannot deal with data streams are because it is difficult for them to maintain the data distribution in the local neighborhood and in finding the KNN in the data stream. This is an exception for methods that were specially designed to tackle data streams.

Discussing further, in Table 4, we present a comprehensive well-known distance-based outlier detection algorithm. We give a summary of different techniques in terms of their computational complexity (running time and memory consumption), address issues, and their drawbacks. Distance-based methods are widely adopted approach since they have a strong theoretical basis and are computationally effective. However, they are faced with some challenges. One of the critical underlining drawbacks of most distance-based methods is their inability to scale well for very high dimensional data sets [144]. Issues like the curse of dimensionality continue to be an evolving challenge. When the data dimension grows, this influences the descriptive ability of the distance measures and makes it quite tricky to apply indexing techniques to search for the neighbors. In multivariate data sets, computing the distance between data instances can be computationally demanding and consequently resulting in a lack of scalability. Even though researchers have focused on solving these problems, we still believe better algorithms can be designed, which can simultaneously address the problem of both a low memory cost and computational time. To address the issue of quadratic complexity, researchers have focused in proposing several significant algorithms and optimizations techniques such as applying compact data structures [124], [145], using pruning and randomization [132], among the many others. Another challenge worth noting is the inability of distance-based techniques to identify local outliers. Distance-based calculations are often done with respect to global information. For K-nearest neighbor approaches, the dataset plays a vital role in determining the perfect KNN score. From most of the algorithms mentioned, choosing an appropriate threshold when it is required is one of the most complex tasks. Another important thing that also influences the results obtained in these outlier detection processes is the choice of k and in choosing appropriate input parameters.

Furthermore, in terms of detecting outliers in the data streams, the fundamental requirement is related to its computational speed. We believe that designing algorithms that can support fast computation in both single and multiple data streams using distance-based techniques will be an exciting challenge for future directions. For current growing interest research areas like that of big data which demands the computation of more massive data sets, it is imperative to design robust algorithms using distance-based techniques that can scale well with a low computational cost (running time and memory) for large up-to-date real data sets for both batch and stream processes.

D. CLUSTERING-BASED APPROACHES

Clustering-based techniques generally rely on using clustering methods to describe the behavior of the data. To do this, the smaller size clusters that comprise significantly fewer data points than other clusters are labeled as outliers. It is important to note that the clustering methods are different from the outlier detection process. The main aim of clustering methods is to recognize the clusters while outlier detection is to detect outliers. The performance of clustering-based techniques is highly dependent on the effectiveness of the clustering algorithm in capturing the cluster structure of the normal instances [146]. Clustering-based methods are unsupervised since they do not require any prior knowledge. So far, numerous research studies are using clustering-based techniques, and some of them are furnished with mechanisms to minimize the adverse influence of the outliers. Zhang [26], in his work introduced many clustering-based algorithms and divided them into different groups. As most of these clustering-based algorithms have not been proposed within this decade, we deem it unnecessary to repeat them in our work and refer our readers to [26] or the original references of the studies for a detailed introduction of the listed algorithms

TABLE 4. summary of distance-based algorithms.

Algorithm	Running Time	Memory Consumption	Issues addressed	Drawbacks
1.Nested-Loop & Indexed Based (ORCA) [122]	$O(dN^2)$ Where d = dimensions of dataset N = Number of objects in datasets.	Has quadratic complexity; hence memory consumption is relatively high.	Solves the issue of lack of support for datasets with more than two attributes. It also tackles the issue of explicit construction of indexing structures and tries to minimize inputs and outputs I/O's.	For d<2, the computational complexity of the indexed base is high.
2.Cell-Based/ Partitioning-based [123]	$O(N)$, That is, it is linear with respect to N but exponential with respect to d	Not assessed	It uses the tuple-by-tuple technique to solve the quadratic complexity (N^2). It also ensures at most three passes over the dataset and thus by far suitable for d<=4, when compared to [123].	For extremely large and high dimensional datasets, the computational complexity increases. Using the global view in the dataset, this makes it taxing to detect some outliers with complex structures.
3.Vertical Distance-based with Local Pruning (VDBLP) [134]	$O(kN)$ Where k is the dimensionality is much smaller than logN.	Not assessed.	It increases the outlier detection rate by applying pruning and by-neighbor methods, and this thus address the scalability issue and gives it better scalability when compared to nested loop [123].	The method of evaluation is not sufficient; it lacks assessment for the memory size and the effect of dimensionality.
4. DOLPHIN [133]	Near linear time.	Near linear time.	It can simultaneously achieve efficient CPU cost and minimizing the I/O cost when the dataset does not fit in main memory.	The algorithm is not suitable for data stream cases where only one scan is applicable.
5.MIRO [135]	It has a linear execution time with respect to N.	$O(N)$	It reduces the execution time of the nested-loop algorithm using several pruning methods.	It is not extended to large and high-dimensional data sets.
Data Streams				
6. EXACT-Neighborship-based Solution (Exact-N) [139]	Shows improved performance over Abstract-C, but still has high CPU cost for maintaining the exact neighborships between objects.	Consumes more memory space, which in the worst-case scenario for the number of data points in the window might result in quadratic memory requirement.	It preserves the exact neighborship learned from the preceding window to lay off the computational intensity in ~~of~~ processing each window. At each sliding window, it reduces the amount of range query search that is needed.	Due to the requirement for storing all exact neighborships among data points, it thus has higher memory consumption when compared to other methods.
7. EXact+absTRActed Neighborship based solution (Extra-N) [139]	Shows improved performance over Abstract-M	It is linear to the number of data points in the window. However, it consumes more memory space when compared to Abstract M.	It provides the data points direct access to their neighbors and then preserves all the exact identified neighbor-ship in previous windows to achieve a minimum number of range query search. It also does not store all exact neighborship in the window, to attain linear memory consumption.	It needs prior knowledge for computing density-based clusters. It also needs to maintain the lifetime of all data points which consumes memory.
6. Abstracted-Neighborship-Based solution Using	Shows improved performance over Exact-N. However, it	Shows improved performance over Extra-N.	It enhances Abstract-C which is unable to keep identified cluster structures	Despite its improvement, it is still faced with similar issues suffered by Abstract-

TABLE 4. ***(Continued.)*** **Summary of distance-based algorithms.**

Membership (Abstract-M) [139]	tends to use more CPU time when compared to Extra-N.		in their previous window. It minimizing the range query search at each window, which is needed for detecting density-based clusters.	C, that is, the extra range query search at each window.
7.Abstract-C [72], [139]	The computational cost is *O(n).* Which shows improved performance over exact-storm for sliding windows and almost the same for count-based windows.	Shows improved performance over Exact-Storm {O ($W^2/S + W$)	It maintains the compact summary of its neighborships and also keeps data points aware of their neighbor's expiration time like that in [137]. It addresses the time consumption issue of Exact-storm as it does not consume much time in searching for preceding active neighbors in every data point.	Since the memory it requires is profoundly affected by the input data stream. It runs range queries for every data point to determine its core point in the window. This thus consume more memory. It gives rise to an unstable CPU performance for the cluster query class due to its dependency on the N-core and the number of core objects.
8.Approximate-Storm [136]	Shows improved performance *{O(W)}* over COD and Exact-Storm. Where *W* is the window size.	Similar to Exact-Storm*{O(W)}*, but the algorithm is readily modified to reduce the space required.	It minimizes the memory usage by storing the only portion of safe inliers for the approximated neighbors and not for the preceding neighbors.	Even though it uses less processing time, but the results are not accurate.
9. Exact-Storm [136]	Similar to Abstract-C with *{O(Wlogk)}* *W* is the window size and *k* is the dimension	*O(kW)*	To address the computation cost issue, it does not store the succeeding neighbor's data points since they do not expire before the data point does.	It stores *k* preceding neighbors for a data point without considering that it might have a large number of succeeding neighbors. Hence, it renders it not good in memory usage, since retrieving preceding active neighbors takes extra CPU time.
10. Continuous outlier detection (COD) [140]	Shows improved performance over ACOD and Abstract-C.	Shows improved performance over ACOD and needs less space as compared to Abstract-C.	Consumes significantly less storage. It can handle multiple values of k.	It is only suitable for fixed values of radius *R* and can't support varying *R* values.
11. Micro-cluster-based Continuous Outlier detection (MCOD) [140]	Shows improved performance over COD, ACOD, and Abstract-C.	Shows improved performance over COD, ACOD, and Abstract-C.	It reduces the range queries and therefore reduces the number of distance computation.	The quality of the cluster is low due to the more compact representation in the form of micro clusters.
12. Advance Continuous outlier Detection (ACOD) [140]	Shows improved performance over Abstract-C.	Shows improved performance over Abstract-C.	It can support multiple values of k and multiple values of R.	Use only in cases of multiple queries with different R values and not for a single R-value.
13.Direct Update of Events (DUE) and Lazy Update of Events (LUE) [140], [73]	DUE shows improved performance over LUE*{O(WlogW)}*	DUE still shows improved performance over LUE*{O(kW)}*, but LUE uses *O(nk)* space.	They effectively re-evaluate the data points that are inliers with the window slides.	It requires extra memory and CPU time to maintain the sorted points.
14.Thresh_Leap [73][142].	$O(W^2 logS/S)$ Where S is the slide size	$O(W^2/S)$	The use of minimal probing rule and the smaller index structure per slide to carry out range queries address the CPU time usage .	It suffers from memory inadequacy when the slide size is small.

TABLE 5. summary of ensemble-based algorithms.

Methods	Performance	Issue/s Addressed	Drawbacks
Lazarevic et al. [37]	The proposed combined technique, when compared with single outlier detection algorithms performance, showed better detection results.	They addressed the issue of effectively detecting outliers in high-dimensional and noisy data.	Although the algorithm showed improved performance for outlier detection task, however, it did not show the performance of the algorithm for large-scale and high dimensional datasets.
Nguyen et al. [38]	The proposed framework is shown to be effective for outlier detection in real-world settings.	They address the problem of ensemble outlier detection in high dimensional databases. They improved on the outlier detection accuracy by combining non-compatible methods of different kinds.	Although the framework covers more studies, it can be expanded using different base outlier detection methods for massive and high-dimensional databases.
Zimek et al. [176]	The outlier detector based on a subsample shows improved performance analytically over individual ensemble methods.	They demonstrated both theoretically and through experiments that combining the individual ensemble, the scores of the outlier members can enhance the robustness and improved the performance of the method.	The approach lacks a better comparison since the sample-based ensemble methods, and feature bagging are not stern competitors.
Zimek et al. [180]	The rank accumulation method shows that it is more suitable for outlier rank combination than its competitor [37]. However, there is no distinct superiority above its competitor.	Address the issue among outlier methods for creating a variety of models and essential ways of combining the outlier rankings.	The scores of different techniques and parametrizations are difficult for the same method because they can vary extensively in magnitude. This results in a lack of fair comparison, and thus, a fair comparison is difficult to achieve for this kind of method.
Zhao et al. [227]	It shows steady improvement when compared to prevailing static combination methods for mining outliers.	They addressed the issue of choosing and combining the outlier scores when there is no ground truth applied for the different base detectors in outlier ensembles.	The method used to describe the k nearest training objects of the test instance as the local region is not ideal. Averaging and maximization are the only ground truth generation methods applied. It lacks more complex and accurate methods can be used.
Zhao et al. [228]	The experimental results show it is superior to its competitors.	They solved the problem of defining the local region and proposed a consistent way of choosing suitable based detectors for the model combination.	Although the method shows improved performance, it can be enhanced with heterogeneous base detectors. it lacks a more consistent ground truth generation approach.

below. Clustering-based outlier detection algorithms have been grouped into the following subgroups.

i. *Partitioning Clustering methods:* are also known as distance-based clustering algorithms. Here, the number of clusters are either randomly chosen or initially given. Some examples of algorithms that fall under this group include PAM [147], CLARANS [148], K-Means [149], CLARA [147], etc.

ii. *Hierarchical Clustering methods:* They partition the set of objects into groups of different levels and form a tree-like structure. To group into different levels, they usually require the maximum number of clusters. Some examples include the MST [150], CURE [151], CHAMELEON [152]

iii. *Density-based clustering methods:* They do not require the number of clusters to be initially given as in the case of partitioning methods; such as K-Means. Given the radius of the cluster, they can model the clusters into dense regions. Some examples of density clustering methods include DBSCAN [153] and DENCLUE [154].

Other groups include the:

iv. *Grid-based clustering methods:*

STING [94], Wavecluster [155], DCluster [156]

v. *Clustering methods for High-Dimensional Data:* CLIQUE [157], HPStream [158]

In addition to the following algorithms, which have been covered in existing literature [5], [22], [26], [33], a two-phase algorithm called DenStream was proposed by Cao *et al.* [9] and D-Stream by Chen *et al.* [159]. The authors make use of the density clustering-based technique to address the problems of both online and offline outlier detections. In DenStream, the initial phase records the summary information of the data streams, and the latter phase clusters the already summarized data. Outliers are detected by introducing potential micro-cluster outlier to differentiate between the real data points and outliers. The main distinction between the two is weight. If the weight is less than the density threshold of the outlier microcluster, then the microcluster is a real outlier. The algorithm, therefore, removes the outlier micro-clusters. To show the effectiveness of the algorithm, it was evaluated against CluStream [160]. The algorithm's efficiency showed improved performance over that of CluStream in terms of memory since they save snapshots on a disk rather than in memory. However, the method still falls short in some areas, for example, in finding arbitrary shape clusters at multiple levels of granularity and in adapting to dynamic parameters in the data streams. Even though the proposed technique was done in 2006, we still believe some future work can be done to address these issues, since, to the best of our knowledge, the problems continue to exist. In the other technique, D-Stream [159], it is similar to DenStream with an online and offline component, except that it is a density grid-based clustering algorithm. Detecting outliers here is not as difficult as in the previous method due to the introduction of sparse, dense and sporadic grids that define the noise. Outliers are considered to be grids whose sparse grids are less than the defined density threshold. The algorithm also shows better performance over CluStream in terms of time performance and clustering performance. Since the algorithms in [9] and [159] use damped window models, Ren *et al.* [161] proposed SDstream, an algorithm that uses the sliding window model. Assent *et al.* [162] proposed AnyOut to compute and detect outliers anytime in streaming data quickly. To identify the outliers in the form of constant varying arrival rates at a given time, AnyOut uses ClusTree to build a precise tree structure. The ClusTree is appropriate for anytime clustering.

Elahi *et al.* [163], using k-means, a clustering-based outlier detection technique was proposed for the data stream that splits the data stream into chunks for processing. However, it does not fit well for grouped outliers. The experimental results illustrated that their method achieved a better performance than some existing techniques [141], [164] for discovering significant outliers over the data stream. However, the authors still believe that by integrating distance-based methods more firmly with the clustering algorithm, it can yield a better result. Moreover, finding other ways to assign the outlierness degree to the detected outliers in the data stream is another good research quest to investigate.

In another study, using a similar concept of k-means as in MacQueen *et al.* [149] together with a rule for the weight, the authors proposed a clustering-based framework to detect outliers in changing data streams [165]. They assign a weight to the feature with respect to their relevance. The weighted attributes are significant because they curb the effect of noise attributes in the algorithm process. When this technique is compared to LOF [8], it has less time consumption, shows a higher outlier detection rate, and also a low false alarm rate. Even though the work showed improved performance over the other baseline algorithm (LOF), it falls short in extending the algorithm for real-world data sets and in investigating its effects. Extending the algorithm to address this issue and in designing new scales for the outlierness degree in the data stream can be an exciting future study. Hosein *et al.* [166] proposed a clustering-based technique, which is an advance k-mean incremental algorithm for detecting outliers in big data streams.

In another work, an unsupervised outlier detection scheme that uses both density-based and partitioning-based schemes for streaming data was proposed by Bhosale *et al.* [167]. The main idea is based on partitioning clustering techniques [168], [148], which assign weights (using weighted k-means clustering) to attributes based on their relevance and adaptivity. The technique is incremental and can adapt to the concept of evolution. It has a higher outlier detection rate than [163]. The authors in this study suggested extending the work for mixed and categorical data for future research.

Moshtaghi *et al.* [169] used a clustering approach to propose a model which labels objects outside the cluster boundaries as outliers. The mean and covariance matrix are incrementally updated to observe the fundamental distribution changes in the data stream. Similar to [169], Moshtaghi *et al.* in another work, proposed eTSAD [170], an approach that models streaming data with established elliptical fuzzy rules. The fuzzy parameters of incoming data are updated as in [169]. This helps in the detection of outliers. Salehi *et al.* [171] proposed an ensemble technique for evolving data streams. Instead of modeling and updating the data streams over time, they proposed using ensemble to generate clustering models. The outlierness value of an incoming data point is calculated by utilizing only the applicable set of clustering models. Chenaghlou *et al.* [172] proposed an efficient outlier detection algorithm, where the concept of active clusters is presented for better time and memory efficient outlier detection result. The input data is split into chunks, and for each current arriving data chunk, active clusters are identified. Here, the models of the underlying distributions are also updated. Rizk *et al.* [173] proposed an optimized calculation algorithm that enhances the process of searching for outliers in both large and small clusters. Chenaghlou *et al.* [174] extend their work in [172] by proposing an algorithm that can detect the outliers in real-time. The algorithm detects outliers in real time and also discovers the sequential evolution of the clusters.

Yin *et al.* [175], proposed some new and effective methods in the context of cluster text outlier detection. In their approach, documents with a low prospect of identifying an existing cluster are referred to as outliers. They conclude that the clusters that possess only one document in the result of Gibbs Sampling of Dirichlet Process Multinomial Mixture (GSDPMM) are classified as outliers in the data set. Since GSDPMM has a great potential for incremental clustering, how to relate GSDPMM in incremental clustering will serve to be an interesting research direction for future work.

When designing clustering-based algorithms for outlier detection, usually the following questions are taking into consideration.

i. Whether the object belongs to a cluster or not, and whether the objects outside the cluster can be regarded as an outlier.
ii. Whether the distance between the cluster and the object is distant or closer. If it is at a distant, can it be regarded as an outlier?
iii. Whether the object belongs to an insignificant smaller or sparse cluster, and how to label the objects within the cluster?

1) CLUSTERING-BASED APPROACHES-ADVANTAGES, DISADVANTAGES, CHALLENGES, AND GAPS

Advantages:

i. They are unsupervised methods which make them suitable choice, and very useful for outlier detection in data streams. After learning from the clusters, additional new points can be inserted and then tested for outliers. This makes them adaptable to an incremental mode. Also, since no prior knowledge is required for the data distribution, this makes it more suitable for incremental mode.
ii. They are robust to different data types. The hierarchical based methods are versatile, they maintain a good performance on data sets containing non-isotropic clusters and also produce multiple nested partitions that give users the option to choose different portions according to their similarity level.
iii. In partitioning cluster related techniques, they are said to be relatively simple and scalable, and thus qualify them for datasets with compact spherical clusters that are well-separated.

Disadvantages, Challenges, And Gaps

i. In clustering settings, outliers are binary; that is, there is no quantitative indication of the object's outlierness. They are also known for their lack of back-tracking ability; therefore, they can never undo what has already been done.
ii. Most clustering methods rely and depend on the users to specify the number of clusters in advance, which is a difficult task. In clustering methods, arbitrary shape clusters also cause some difficulties in realizing the exact clusters of the data. Therefore, most existing clustering algorithms require several, in advance and the shape of the clusters to be defined. However, in data stream scenario to assume several clusters in advance is very daunting.
iii. Partitioning methods are said to be highly sensitive to the initialization phase, outliers, and noise. Similar to the density-based clustering techniques, they also carry some setbacks with regard to the fact that they are highly sensitive to the setting of the input parameters. They have inadequate cluster descriptors and mostly unsuitable for very large high-dimensional datasets because of the curse of dimensionality. Furthermore, in some of the hierarchical methods [150], [152], the cost of clustering is enormous for high dimensional and massive datasets. The vagueness in the criteria for termination and the severe dilapidation of the effectiveness in high dimensional spaces as a result of the curse of dimensionality is another drawback.

Despite the challenges and drawbacks, clustering based techniques are useful in outlier detection, more especially in data streams. Clustering-based algorithm for the process of outlier detection in data streams has drawn the attention of researchers and is seen as an interesting domain. The challenges of choosing appropriate cluster width and calculating the distance between objects in multivariate data are among the obstacles researchers try to solve. The detection rate is high in most cases, but they are also challenged with high false positives. The density clustering-based techniques handle noise while the partitioning and hierarchical methods do not. These techniques have their strength and weaknesses; it is challenging to decide which one is superior to the other. Interesting work in the future will be to choose a suitable dataset and evaluate these methods using different evaluation metrics. Also, a hybrid approach using the pros of different techniques. For instance, density-based clustering methods are suitable to cluster arbitrary shapes. Reducing the computational cost, the speed in the complex and large dataset for partitioning-based methods and hierarchical methods is also another interesting future work. Other interesting future studies will be to propose algorithms to detect outliers in low-density regions or where outliers are within clusters with a small number of data points. In addition, suggesting methods that calculate the distance of the data point to the nearest cluster centroid to detect the outliers efficiently.

From the selected clustering-based techniques mentioned, we can see that not much work has been done recently. To the best of our knowledge, using a thorough analysis of this technique for outlier detection. Therefore, we do not include any summary table of this technique, unlike in the other previous approaches.

E. ENSEMBLE-BASED APPROACHES

Ensemble-based methods are generally used in machine learning as a result of their comparatively better performance when compared to other related methods.

Although ensemble-based techniques for outlier detection when compared to other OD methods have had very few reports [37], [38], [176]–[182]. However, they are often used in recent outlier detection problems [183], [184], and have more open challenges. Ensemble techniques are used in cases where one is prompted to answer the question of whether an outlier should be a linear-model based, distance-based, or other kinds of model-based. They are usually applied in classification and clustering problems. They combine the results from dissimilar models to produce more robust models and then reduce the dependency of one model to a particular dataset or data locality. However, ensemble methods in the context of outlier detection are known to be very difficult. In recent years, several techniques have been introduced, including the following: (I) Bagging [37] and boosting [184] for classification problems (ii) Isolation forest [192] for parallel techniques. (iii) for sequential methods [185] and Extreme Gradient Boosting Outlier Detection (XGBOD) [183] and a Bagged Outlier Representation Ensemble (BORE) [186] for the hybrid methods.

Lazarevic *et al.* [37], proposed the very first known ensemble method on improving outlier detection using the ensemble method. It makes use of the feature bagging approach to handle very large high dimensional datasets. The technique combines the outputs of multiple outlier detection algorithms, each of which is created through a random designated subset of features. Each of the algorithms randomly selects a small subset of its real feature set and then assigns an outlier score. The score is assigned to all the data records that match up with the probability of them being considered outliers. Each of the outlier score obtained from the different algorithms is then combined to get better quality outliers. From their experiment, it shows that the combined method can produce a better outlier detection performance because it focuses on smaller feature projections from the combined multiple outputs and distinct predictions. However, considering how to fully characterize these methods for very large and high dimensional datasets would be motivating future work. Also, examining the impact of shifting the data distributions in detecting the outliers for each round of the combined methods (not limited to only distance-based approaches but other approaches) is worth considering.

Aggarwal [178], presented a study on outlier ensemble analysis, which has recently provoked great interest in literature [187], [236]. He discusses various outlier ensemble methods and how such outlier ensemble analyses can be more effective. He further explained how these methods are connected to ensemble-methods in data mining problems. Some examples of outlier ensembles in the context of classification and clustering were then given. In the classification context, boosting [187] and bagging (Bootstrap Aggregating) [37] are two examples of ensemble-based methods that have been proposed. In the context of clustering, the Multi-view [188] and alterative clustering [189] serve as examples. Another critical study in their work is how to categorize ensemble outlier analysis problems, whether they are independent or sequential ensembles, data-centered, or model-centered ensembles. The ensemble algorithms are classified by the ''component independence'' which tries to answer to the question of whether the different components of the ensemble are independent or dependent on one another. For example, in boosting where the results depend on a prior execution, such a method is not independent of the other, while bagging is the opposite; they are independent of one another. For the ''component type,'' each component of an ensemble is described according to its model choice or data choice. The ''model-centered'' is independent, while the ''data-centered'' is sequential. However, one cannot give an ultimate conclusion because it might depend on the foundation of the data and models.

Other succeeding studies [38], [190], [191] in later years that focused on using ensembles for outlier detection faced numerous challenges. Some of these challenges include the issue of comparing the scores using different functions and mixture models to fit the outlier scores and to give a score combination. In addition, issues of how to support the combination of different detectors or methods to form one ensemble arise. Schubert *et al.* [191], compared the outlier ranking based on the scores using similarity measures. A greedy ensemble technique was proposed as an application, which shows the significance of the performance of ensembles through diversifying approaches. Earlier in 2010, Nguyen *et al.* [38] studied the difficulties of ensemble OD methods for high dimensional datasets. They proposed a unified framework that combines non-compatible methods of different outlier detection algorithms. Instead of applying the same approach each time to determine the outlier score, various detection methods are applied to approximate the outlier score. Using the formal concept of the outlier score, they propose Heterogeneous Detector Ensemble on random Subspaces (HeDES) through the combination of functions, to address the issue of heterogeneity. Unlike the Lazarevic *et al.* [37] framework, HeDES can bring together different techniques that produce different outlier scores and score types; for instance, a real-value against that of the binary-value. Even though from their experimental studies, the framework shows effectiveness in the detection of outliers in a real-world data set, we believe considering an orderly extension in doing a further experiment on all possible combined functions. In addition, extending the analysis to larger and higher dimensional datasets could be interesting future work.

Zimek *et al.* [176] proposed a random subsampling technique to estimate the nearest neighbors and then its local density. Usually, applying subsampling techniques from a set of given datasets, it will obtain the training objects without replacement. This can improve and enhance the outlier detection method performance. Using other outlier detection algorithms coupled with a subsampling technique can give a different set of results and higher efficiency.

Zimek *et al.* [180] in another work, the authors considered their technique from the perspective of learning theory

as another possible approach to ensemble outlier detection. To construct the outlier detection ensemble, the authors proposed a data perturbation technique that brings forth diversity in different outlier detectors and a method that combines distinct outlier rankings. The main focus of their approach utilizes the notion of distance and density estimations in Euclidean distance type dataspaces. To get a more consistent density estimate, the attribute values at each point are altered by adding small randomized amounts of noise. All the i perturbed bootstrapped data sets then go through a selected outlier detection algorithm, which helps in recording each data point identity, aggregates the scores and then ranks the positions. The i outlier scoring (or rankings) are then combined to attain a steadier and dependable outlier scoring of the data.

Pasillas-Diaz *et al.* [177] considered both subsampling and feature bagging techniques. The feature bagging technique is used to obtain the various elements at each iteration, while the subsampling technique calculates the outlier scores of the different subsets of data. One key drawback in their method is the difficulty in obtaining the variance of the objects through feature bagging. Also, the size of the subsampled dataset influences the sensitivity of the final result.

Zhao *et al.* [227] proposed Dynamic Combination of Detector Scores for Outlier Ensembles (DCSO) an unsupervised outlier detector framework. DCSO tries to solve the challenge of choosing and combining the outlier scores in the absence of the ground truth for different base detectors. It selects the most suitable base detectors, with focus on the locality of the data. DCSO initially labels the local region of a test instance with respect to its k nearest neighbors. It then detects the base detectors that show the best performance within the local region. Zhao *et al.* [228] proposed Locally Selective Combination in Parallel Outlier Ensembles (LSCP) framework to address the same issues in [227]. They use a similar approach as in [227] and presented four variations of the LSCP framework.

For more details and broader discussions about outlier ensemble techniques, the Aggarwal *et al.* [229] outlier ensemble book gives detailed discussions on outlier ensemble methods. Although most of the studies mentioned there were done before 2017, however, the book itself is very comprehensive and rich in details for the understanding of outlier ensemble methods. It presents the different types of ensemble methods and categorize them into different types. In addition, it gives an overview of the outlier ensemble design frameworks.

1) ENSEMBLE-BASED APPROACHES-ADVANTAGES, DISADVANTAGES, CHALLENGES, AND GAPS

Advantages

i. They are more stable and give better predictive models. The availability of algorithms like boosting and bagging enhances the ensemble methods to perform more efficiently. They improve the robustness in the data mining process by minimizing the dependence of the model on a particular data set.
ii. They are suitable for outlier analysis in high dimensional data; for example, Lazarevic *et al.* [37] applied feature bagging for outlier detection in high dimensional data.
iii. In noisy and streaming scenarios where an individual classifier's result is not very robust due to the processing time and data quality problems, ensemble analysis is instrumental.

Disadvantages, Challenges, And Gaps

i. Ensemble techniques in the context of detecting outliers when compared to other data mining problems are poorly developed. This is as a result of the difficulties in evaluating the features of the ensembles. Moreover, selecting the right meta-detectors is a difficult task.
ii. For real datasets, the outlier analysis can be very complex to evaluate due to the combination of a smaller sample space and its unsupervised nature. This can further result in the incorrect prediction of the steps of the algorithm in making robust decisions without triggering the over-fitting problem.

Although outlier ensemble techniques have shown promising results, they still have areas for improvement. Ensemble analysis techniques can be very useful in areas where the data is noisy and in streaming scenarios. This is mainly because in these scenarios they are usually challenged with some drawbacks, such as the quality of the data and the processing time that makes the results produced from individual classifiers not very robust. More techniques are being proposed to address the challenge of model combinations.

To address these challenges and many others proposed by Zimek *et al.* [181], several additional methods have been proposed [38], [182], [190]–[192] to improve outlier detection using ensemble methods, but most of these methods are meta methods except for those suggested by [37]. To further discuss and delve deep into outlier ensembles techniques, Zimek *et al.* [181] in their study have presented several open questions and challenges in using ensemble methods in the detection of outliers. Although some new emerging research work has started to contribute to these open research problems [180] however, topics about the issues of proposing diversifying principles and how to combine outlier rankings remains to be open and engaging for future research directions. Some techniques [181], [184] are static and do not involve any detector selection methods. This kind of technique [184] that is characterized by the absence of a detector selection process hinders the performance of the manner in identifying the unknown outlier cases. Another significant aspect that is not given much attention is the importance of data locality. An open research problem will be to consider the data locality. That is, instead of only focusing on evaluating the competence of the base detector on a more global view, the local region with respect to the test objects can be considered as well. This will help in the detector selection and

combination processes. Other essential problems for further research, are in addressing the issue among ensemble outlier methods for creating a variety of models and meaningful ways of combining the outlier rankings.

F. LEARNING-BASED APPROACHES

Learning-based methods for the process of outlier detection have been applied in different sub-disciplines in machine learning - in active learning, graph-based learning, and deep learning. In the subsequent section, we will introduce some research in outlier detection that make use of these learning methods.

1) ACTIVE LEARNING

Active learning is an example of a semi-supervised learning method in which designed algorithms interact with the user or information source to get the desired outputs [193], [194]. For example, in cases of some real dataset with huge unlabeled datasets, the task of manually labeling these data is expensive. Such a scenario demands the learning algorithm to query the information source or user actively. When applying an active learning algorithm in such a scenario, the algorithm will be able to discover those smaller fractions of instances that were labeled by the user in the training data set. This is done to boost the improvement of the re-trained model. Active learning resembles a system in which the learning algorithm can request the user for input labels of the instances to give better predictions. Active learning for outlier detection has recently been embraced in different research domain [195]–[199]. Aggarwal *et al.* [6] use the concept of active learning in outlier detection to solve the ambiguity of giving clear reasons why outliers are flagged and what prompts the relatively high computational demand for density-estimation based OD methods. In their approach, they initially apply the classification techniques to the labeled dataset that contains potential outliers (artificially generated). The active learning method is then applied to minimize the classification problem through a selective sampling mechanism known as ''ensemble-based minimum margin active learning.'' Gornitz *et al.* [200] proposed another work where an active learning strategy is applied for anomaly detection. To obtain a good predictive performance, they repeat the process of alternating between the active learning process and the update of the model. The active learning rule is applied after the method is trained on unlabeled and improved examples.

Das *et al.* [196], [197] used an active approach to query the human analyst to obtain a better result. They avidly select the best data instances for the querying process, but no clear insight, explanation, and interpretation of the design choice were given. In the next study, they try to address these issues. In 2019, Das *et al.* [201] then proposed an active outlier detection method via ensembles called GLocalized Anomaly Detection (GLAD). They study how to automatically fit ensemble outlier detectors in active learning problems. In GLAD, the end-users maintain the use of modest and comprehensible global outlier detectors. This is attained through learning automatically their local weight in particular data instances by means of label feedback. The fine-tuning of the ensemble helps in maximizing the number of correct outliers discovered. This kind of framework is referred to as a human-in-the-loop because the human analyst for each round of iterations gives label feedback.

Even though active learning for outlier detection has recently been embraced in the research domain, they still lack in the literature, and there is still more work that needs to be done. The process of discovering true outliers by the human analyst can be difficult, the need for the techniques to minimize the effect of false positives through the design and configuration of an effective outlier detector is needed for the human analyst in the future. In addition, better insights and interpretations of outlier scores and related results obtained through employing different algorithms are needed. Active learning in the context of outlier detection needs solid interpretations and explanations for it to be well understood in the research community. Finally, the design of active learning algorithms for handling data streams is also a promising research challenge.

2) SUBSPACE LEARNING

Outlier detection methods mentioned to this point usually detect outliers from the complete data space considering all the dimensions. But most outliers are often denoted as rare neighborhood activities in a declining dimensional subspace. For objects with several attributes, Zimek *et al.* [179] denote that, only subsets with important attributes give valuable information. While characteristics like the residual attributes contribute little or no importance to the task and might delay the process of separating the OD model in solving such an issue, it will be interesting to perceive the outliers from a suitable subspace.

In the outlier detection field, subspace learning is widely studied and applied in high dimensional problems. For subspace learning-based OD approaches, the main objective is to discover meaningful outliers in a well-organized way by examining dissimilar subsets of dimensions in the dataset. Mostly, these approaches are divided into sparse subspace [195], [196] and relevant subspace [126], [198], [202] methods. The former project the high-dimensional data points onto sparse and low dimensional subspaces. These objects within the sparse subspace can then be labeled as outliers since they are characterized with a lower density. One big drawback of these methods is the time consumption with regards to exploring the sparse projections from the entire high-dimensional space. To address this drawback, Aggarwal *et al.* [6] proposed a method that improves the effectiveness of exploring the subspaces. The subspaces are achieved through an evolutionary algorithm. However, the performance evaluation of the algorithm is highly dependent on the initial population.

An additional method that focuses on the path of the sparse subspace approaches is the Zhang *et al.* [195] method.

Here, the concept of the lattice is used to signify the subspace relationships. The sparse subspace here also is those with low-density coefficients. Again, creating the idea of lattice influences and hinders the efficiency of the method because of its complexity, and this results in low efficiency. Dutta *et al.* [196] proposed a way to achieve sparse space. They applied sparse encoding to develop objects to multiple linear transformation space. The OD method uses relevant subspaces to examine the local information, which is useful for identifying outliers since they are essential features. Huang *et al.* [126] proposed Subspace Outlier Detection (SOD), a kind of relevant subspace method. Here, every objects correlation with its shared nearest neighbors is examined. They use the ranks of individual objects that are close as the degree of proximity of the object, but not take into consideration the objects distance information with respect to their neighbors. SOD focuses mainly on the variances of the features. Muller *et al.* [202] proposed another method to determine the subspaces. They make use of the significant relationships of the features; unlike SOD that only focuses on the variances of the features. However, a significant drawback of their method is its computational demand.

In a similar study, Kriegel *et al.* [17] applied principal component analysis to get the relevant subspaces and Mahalanobis distance computation through gamma distribution to detect the outliers. The key difference compared to the previous study [202], is that a large amount of local data is needed to identify the deviation trend. This consequently affects the flexibility and scalability of the method. To address the issue of flexibility of the technique, Keller *et al.* [85] designed a flexible OD technique which uses subspace searching and outlier ranking process. Initially, using the Monte Carlo sampling method, they obtained the High Contrast Subspaces (HiCS) and then combined the LOF scores based on the HiCS. Stein *et al.* [203] then proposed a local subspace OD method by adopting global neighborhoods in another study. Initially, the HiCS obtained all the relevant subspaces and instead of LOF, and LoOP technique was used to calculate the outlier scores.

Although subspace learning OD methods show high efficiency and are useful in some cases, they are generally computationally expensive. This is because, in subspace learning methods, there is a prerequisite in exploring the subspace high dimensional space. Discovering the relevant subspaces for the outliers can also be another difficult task. Designing and proposing effective methods to handle these challenges can be exciting research in the future of subspace OD related methods.

3) GRAPH-BASED LEARNING METHODS

The use of graph data is becoming universal in many domains. Using graph-based learning for OD methods has been the focus of some researchers. In graphs, objects usually take the form of long-range connections, and a set of new techniques has been proposed for outlier detection in graph data. Akoglu *et al.* [34], presented a comprehensive survey of graph-based outlier detection techniques and descriptions. They included state-of-the-art methods and some open research challenges and questions. Furthermore, the importance of adopting graphs for outlier detection techniques was given. The graph-based approach in outlier detection is vital as they show the inter-dependent state of the data, show insightful representations, and robust machinery.

Moonesinghe *et al.* [204] proposed Outrank, which is among the first constructed graph-based outlier detection framework. From the original data set, they developed fully linked undirected graphs and applied the Markov random walk method on the predefined graph. The stationary distribution values of the random walk serve as the outlier scores. In the most recent study, Wang *et al.* [205] proposed a novel method that combines the representation of the graph together with each object's local information in its surroundings. They address the problem of a high false-positive rates in the OD methods, which is usually as a result of the neglect of the local information around each node for graph-based methods. The local information obtained from around each object helps in the construction of a local information graph. The outliers are detected by computing the outlier scores through the process of a random walk on the graph. Wang *et al.* [206] in another study proposed another OD method that captures different local information from different standpoints. They used multiple neighborhood graphs, and the outlier scores are deduced through a random walk on the predetermined graph. These methods all show improved performances as claimed by the authors. However, since using graph-based learning methods has not yet been widely embraced, it is another domain for outlier detection research in the future.

4) DEEP LEARNING METHODS

Recently, more attention has been given to deep learning in many areas including several studies related to outlier detection problems [35], [36] [30], [207]–[209]. Most recently, Chalapathy and Chawla [32] in their survey presented a comprehensive study of deep learning methods for outlier detection. They review how deep learning methods are used in various outlier detection applications and evaluate their effectiveness. The use of deep learning techniques in detecting outliers is important because of one or several of these reasons. (1) the need for better ways of detecting outliers in large-scale data. (2) better ways of learning the hierarchical discriminative features from the data (3) better ways to set the boundary between a normal and unusual behavior in continuous evolving data sets. Deep learning can be based on supervised, semi-supervised, and unsupervised approaches to learning data representations. For example, employing the deep learning concept in fraud and anti-money laundering systems can detect and identify the relationships within the data, and subsequently enable researchers to learn the data points that are not similar to each other and then predict outliers.

In the supervised deep OD methods, the binary or multiclass classifier are trained by utilizing the labels of the

normal and abnormal data instances. The supervised models, for instance, that are framed as multi-class classifiers, help in identifying abnormal behaviors such as fraudulent health-care transactions [32]. Although supervised methods are shown to have improved performance, however, semi-supervised and unsupervised methods are mostly adopted. This is because, supervised methods lack the readiness of labeled training data and also, there is a problem of class imbalance, which makes it sub-optimal to the others.

In semi-supervised deep OD methods, the ease of obtaining the labels of the normal instances compared to the outliers makes it more widely appealing. They make good use of prevailing normal positive classes to differentiate the outliers. Semi-supervised techniques can be applied for training deep autoencoders on data samples missing outliers. With enough normal class training samples, the autoencoders will show significant improvement for the normal instance, with fewer reconstruction errors over the abnormal event.

In unsupervised deep OD methods, the outliers are detected exclusively on the essential features of the data instances. Here, the data samples are unlabeled, and unsupervised OD techniques are used to label the unlabeled data samples. In most of unsupervised deep OD models, autoencoders play a central role [210], [238]. Most emerging research studies adopting deep learning techniques for OD methods utilize unsupervised methods. Using deep learning for unsupervised outlier detection problems has shown to be effective [211], [212]. They are mostly categorized into the model architecture adopting autoencoders [213] and hybrid models [214]. The autoencoder related models assess the anomalies through reconstruction errors, i.e., employing the magnitude of the residual vector, whereas, in the later models, the autoencoder is used as the feature extractor, and then the hidden layers represent the input. In another study for deep learning models, Dan *et al.* [215] proposed Outlier Exposure (OE) to improve the outlier detection performance. They offered a method through iterations to find a suitable classifier for the model to learn the heuristics. This helps in differentiating between the outliers and in-distribution sample.

In another study, Du *et al.* [216] proposed Deeplog, a universal framework that adopts a deep neural network approach for online log outlier detection and analysis. The deeplog utilizes Long Short-Term Memory (LSTM) to model the system log. The whole log messages are learned and encoded by the deeplog. Here, the anomaly detection process is done for every log entry level, in contrast to other methods per session level approach. Borghesi *et al.* [217] proposed a new way of detecting anomalies in High-Performance Computing Systems (HPCS) through adopting a kind of neural network called autoencoder. They first choose a set of autoencoders and train them to learn the normal pattern of the supercomputer nodes. After the training phase, they are applied to identify abnormal conditions.

In deep OD methods, based on the training objectives, these methods can employ Deep Hybrid Models (DHM) or One Class Neural Networks (OCNN) [32]. The DHM uses deep neural networks. It focuses primarily on autoencoders for feature extraction, and the hidden representation of the autoencoders learned serves as the input for detecting outliers for most OD algorithms such a One-Class SVM. Although hybrid approaches maximize the detection performance of outliers, however, a notable limitation is the shortage of trainable objective solely designed for outlier detection. Therefore, DHM is limited in extracting rich differential features to detect the outliers. To solve this drawback, Chalapathy *et al.* [218] and Ruff *et al.* [219] proposed One class neural networks and Deep one-class classification, respectively. The One-Class Neural Networks (OC-NN) combines the advantage of deep networks ability to extract rich feature representations of the data and the benefit of one-class creating a close-fitting structure around the normal data.

The deep learning OD based technique is still active to be explored further and are promising for future work. In the discussion section, we propose and recommend some open challenges for future research work.

5) LEARNING-BASED APPROACHES-ADVANTAGES, DISADVANTAGES, CHALLENGES, AND GAPS

a: ADVANTAGES

In OD based learning methods, such as in active learning, the time-consumption in detecting outliers it reduced since the technique is not passive learning. It helps to reduce the number of labeled data needed for training the model to discover the outliers. Graph-based methods show the vital inter-dependent state of the data and provide an insightful representation for detecting the outliers. The deep learning techniques help in delivering and showing better ways of learning the hierarchical discriminative features from the data. They provide better ways of detecting outliers in large-scale data. In addition, they offer better ways to set the boundary between normal and unusual behavior in continuous evolving data sets.

b: DISADVANTAGES, CHALLENGES, AND GAPS

Some learning-based techniques such as subspace learning, can be computationally expensive and challenging to discover the relevant subspaces for the outliers. In areas like deep learning techniques, with an increase in the volume of data, it becomes a big challenge for the possibility of traditional methods to scale well to detect outliers. The need to design deep learning OD techniques to capture complex structures in large-scale data is crucial. In addition, the traditional manual learning process to extract features from the data has many disadvantages. Therefore, finding better ways through learning the hierarchical discriminative features from the data is vital. The lack of accurate representation of normal and abnormal boundaries also presents challenges for both traditional methods and deep learning-based methods. Addressing these challenges can be interesting work in the future. There are still limited studies using unsupervised algorithms such as Long Short-Term Memory networks (LSTM), Recurrent

Neural Network (RNN), Deep Belief Network (DBN), etc. in the area of outlier detection. For in-depth knowledge and more references, we suggest Chalapathy *et al.* [32] and Kwon *et al.* [30] surveys.

IV. EVALUATION TECHNIQUES, TOOLS, AND DATASETS FOR OUTLIER DETECTION PROBLEMS

A. EVALUATION METHODS

Many outlier detection algorithms have been proposed over the years, but one major challenge in data mining research is how to evaluate these methods. Different techniques have been proposed to tackle this problem. Over the years, with the increasing flow of outlier detection algorithms, some researchers have claimed that their method outperforms other methods without a thorough analysis from a broader perspective. Therefore, researchers have seen this as an open research direction to find ways of evaluating different algorithms. In recent years, some research studies have concentrated on the evaluation of OD methods, for example, for distance-based, ensemble-based approaches, and unsupervised methods.

Distance-based methods over the last decade have a considerable amount of literature, and evaluating these methods is very significant. However, there are many challenges in assessing these techniques against each other. The main interest of most researchers and practitioners in outlier detection problems is the effectiveness and efficiency. For example, in terms of the efficiency, evaluating the efficiency of different methods can be quite complicated because the performance will depend on factors such as the size of the data, the dataset dimensionality, parameter choice, and other details related to the implementation.

Orair *et al.* [220], focused on evaluating several distance-based methods for outlier detection approaches to infer some useful guidelines in designing optimized outlier detection algorithms. An outlier detection framework was implemented, and a factorial experiment was conducted on a couple of the optimization strategies that have been proposed in recent times. It is done to evaluate the pros and cons of these optimization techniques. The results obtained from the factorial experiment is beneficial for giving significant and interesting insights. For instance, they found that certain combinations of optimizations work efficiently for real and synthetic data sets. However, none of the combinations of the optimization can claim to be superior to the other in all types of data. The authors proposed three optimization techniques

i. *Approximate Nearest Neighbor Search (ANNS)*

ii. *Pruning:* is the preprocessing step used by several algorithms to facilitate the partitioning or clustering of data points. The pruning scheme proposed by the authors are:
- *Pruning partitions during the search for Neighbors (PPSN).*
- *Pruning partitions during the search for outliers (PPSO).*

iii. *Ranking:* The main objective is to improve the efficiency of ANNS pruning rule. There are two sub-categories of optimization strategies, which include:
- *Ranking Objects Candidates for Neighbors (ROCN)*
- *Ranking Objects Candidates for the Outlier (ROCO)*

The authors classified their work according to whether these algorithms go through clustering preprocessing phase, the type of pruning scheme use, and whether an object's candidates can be ranked according to neighbors and outliers. From their study, it is apparent that one cannot justify the effectiveness of any single optimization or the combination of optimizations over the other, as it always relies on the characteristics of the dataset. In another study, Achtert *et al.* [43] propose a visualization tool [221], [222] to compare and evaluate outlier scores for high dimensional data sets. Over the years, many approaches have presented the degree of an object being considered as an outlier through an outlier score or factor. However, these outlier scores or factors vary in their contrast, range, and definition among various outlier models. This makes it quite difficult for a novice user with OD methods to be able to interpret the outlier score or factor. In some cases, even for the same or similar outlier model, the same score within one or in the same data set can depict a different outlier degree. For illustration, the same outlier score x in database y and database z can have a considerable degree of outlierness. This makes it also very tedious to interpret and to compare the outlier scores. In addition, we should consider that in different models, various assumptions are made, and therefore, this might directly influence the interpretation of the degree of outlierness in the data set and how to define an outlier in the same datasets or for different datasets.

Not much provision for better evaluation techniques has been proposed in recent studies, and most studies concentrate on introducing new methods to improve the detection rate and computational time. In contrast to classification problems, the evaluation of outlier detection algorithms performance is more complicated. Researchers have provided several adopted measurements to evaluate outlier detection algorithm performance [223]. They are defined as follows:

i. *Precision* - this denotes the ratio of the number of correct outliers m, divided by the whole number of outliers t. In a particular application, setting t can be difficult. Therefore, t is usually assigned as the number of outliers in the ground truth.

ii. *R-precision* - this refers to the proportion of correct outliers in the top number of ground truth potential outliers identified. The R-precision does not contain enough information because the number of true outliers is minimal when compared to the total size of the data.

iii. *Average precision* - this denotes to the average of the precision scores over the ranks of the outlier points. It combines recall and precision.

iv. *Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC)* - the ROC is a graphical plot that shows the true positive rate against the false positive rate. The true or false positive rate signifies the number of outliers or inliers ranked among the potential outliers in the top number of outliers in the ground truth. The AUC shows the numerical evaluation performance of the outlier detection method.

v. *Correlation coefficient* - is a numerical measure of correlation, i.e., a statistical relationship between two variables. For instance, Spearman's rank similarity or Pearson correlation. More importance is placed on the possible outliers ranked at the top.

vi. *Rank power (RP)*-it ranks the true outliers at the top and normal ones at the bottom. It comprehensively evaluates the ranking of true outliers.

Most of the evaluation methods are instead heuristic and focus on precision, the receiver operating characteristic (ROC) curve and area under the curve (AUC) in showing the results. The drawback of these evaluation procedures is that there is no provision for a similarity check among the methods. Knowing how similar or correlated the ranking of outlier scores are is considered a very significant step towards constructing better OD methods. AUC completely disregards the small variations among scores and only considers the ranking. It is also inferior and not perfect for unbalanced class problems when compared to techniques such as the area under the precision-recall curve, which shows a better possibility in highlighting small detection changes. However, despite these drawbacks, AUC, ROC, and precision-recall still serve as the de facto standard in evaluating many outlier detection problems. Since knowing how similar or correlated the ranking of outlier scores are is a very significant step towards constructing better OD methods, Schubert *et al.* [191] in their study gave a global view that permits the evaluation of the performance of different approaches against each other. The proposed framework considers the problem of class imbalance and then offers a new understanding about the similarity and redundancy of prevailing outlier detection techniques. To achieve the main objective in giving a better evaluation for both the outlier rankings and scores, a suitable correlation measure for comparing rankings by taking into account the outlier scores was established.

In another study, Goldstein *et al.* [102] proposed a comparatively universal evaluation of nineteen different unsupervised outlier detection algorithms with ten publicly available datasets. The main aim was to address the lack of interesting literature that exists [224], [225] that gives a better evaluation of outlier detection algorithms. One notable trend with existing research literature is the comparison of newly proposed algorithms with some previous or state-of-the-art methods. However, most of these studies fail to publish the datasets with appropriate preprocessing or indicate which application scenarios they are most suitable. They also mostly lack a clear understanding of the effect of the parameter k and the established criteria of whether it is a local or global outlier. The authors address these issues by performing an evaluation study that reveals the performance of the effect of the parameter settings, computational strength, and the overall strengths and weaknesses of different algorithms. The list of algorithms was categorized into nearest-neighbor based methods, statistical based methods, clustering based methods, subspace methods, and classifier-based techniques. These algorithms were then compared. For the KNN methods, the choice of the parameter k is very significant as it influences the outlier score. Other important things to consider are the dataset, the dimensionality, and normalizations. They experimented with investigating the influence of the parameter and then evaluated the nearest neighbor algorithms. Key findings from their study were that local outlier detection algorithms such as LOF [8], INFLO [75], COF [80] and LoOP [81] are not suitable for detecting global outliers since they showed poor performance on datasets comprised of global outliers. However, it is the opposite of the performance of global outlier detection problems on local outlier detection problems. In addition, they found that the clustering-based algorithms were in most cases inferior to the nearest-neighbor based algorithms. Therefore, it is recommended for a global task to apply the nearest-neighbor techniques, and for a local task, the local outlier algorithms like LOF are more suitable than other clustering-based methods.

Another issue with regard evaluating most OD models is that there is a scarcity of rich knowledge about the strength and weaknesses of these outlier detection models, suitable benchmark datasets for outlier detection task and some biases that are used in the evaluation process that are not well understood. Campos *et al.* [226], similar to [97], did an experimental study across a wide variety of specific datasets to observe the performance of different unsupervised outlier detection algorithms. In their study, they classified different datasets and deliberated on how suitable they are as outlier detection standard datasets. Also, they further discuss and examine the commonly-known and used methods/measures for comparing outlier detection performance. Some common misconceptions the authors clarify are, for instance, the ground-truth datasets containing a large number of outliers. It is sometimes believed that these outliers will influence the evaluation performance of these methods, but this does not hold in all scenarios. Usually, the large proportions of outliers in datasets are not suitable to evaluate outlier detection techniques because outliers are supposed to be less common in the datasets. A small percentage of outliers and normalized datasets usually produce a much better performance in most cases. Another critical misconception held is concerning the influence of the dimensionality. An increase in the dimensionality often results in a high computational cost but is not directly proportional to the overall performance, especially in terms of the detection rate.

Another important area to consider is the evaluation of outliers in data streams. Outlier detection in data streams is usually a difficult task, because the data should be learned and processed in real-time while concurrently making good

predictions. Except for the Lavin *et al.* [230] Numenta Anomaly Benchmark (NAB) framework, there is a lack of benchmarks to effectively test and score the effectiveness of real-time outlier detection methods. With more recent studies concentrated in this domain, there is a need for proposing efficient and rigorous benchmarks to evaluate real-time outlier detection algorithms in data streams effectively.

B. TOOLS FOR OUTLIER DETECTION

In outlier detection, many tools and datasets have been used. Here, we introduce some popular tools used for outlier detection processes and some outlier detection databases.

The prevalence of outlier detection in industrial applications has seen the development of many software tools such as the following provided below.

Scikit-learn Outlier Detection [231]

The scikit-learn project offers some machine learning tools that can be applied for outlier detection problems. It includes some algorithms like LOF [8] and Isolation Forest [192].

2) *Python Outlier Detection (PyOD)* [232]

PyOD is used for detecting outliers in multivariate data. It is a scalable python tool that has been used in many research and commercial projects, including new deep learning and outlier ensembles models [60], [62], [233].

3) *Environment for Developing KDD-Applications Supported by Index-Structures (ELKI)* [43]

ELKI is an open source data mining algorithm that provides a collection of data mining algorithms, including OD algorithms. It allows the ease and fair assessment and benchmarking of OD algorithms. It is written in Java.

4) *Rapid Miner* [234]

The extension of this tool contains many popular unsupervised outlier detection algorithms such as LOF, COF [80], LOCI [82], and LoOP [81].

5) MATLAB [235]

MATLAB also supports many outlier detection algorithms and functions. Algorithms can be implemented using MATLAB because it is user-friendly.

6) *Massive Online Analysis (MOA) tool* [143].

MOA is an open source framework that provides a collection of data stream mining algorithm. It includes some distance-based outlier detection algorithms such as COD, ACOD, Abstract C, MCOD, and some tools for evaluation.

C. DATASETS FOR OUTLIER DETECTION

Outlier detection methods have been applied in different kinds of data, such as in regular and high-dimensional data sets [240], streaming datasets, network data, uncertain data [241], and time series data. In outlier detection literature, two types of data are mostly considered and required for evaluating the performance of the algorithms. They are real-world datasets and synthetic datasets. The real-world datasets can be obtained from publicly available databases. Some of the most popular and useful databases that contain real-world datasets for outlier detection include the following:

1) *The UCI repository* [52].

The UCI repository has hundreds of freely available data sets, and many OD methods use the repository to evaluate the performance of the algorithms. However, the majority of these datasets are designed for classification methods. In outlier detection scenarios, the generally used approach is to preprocess the datasets. The outliers represent objects in the minor class, and the rest are considered as the normal ones.

2) *Outlier Detection Datasets (ODDS)* [51].

Unlike UCI repository, ODDS, provides open access to a collection of datasets only suitable for the outlier detection process. The datasets are grouped into different types including multi-dimensional datasets, time series univariate and multivariate datasets, and time series graph datasets.

3) *ELKI Outlier Datasets* [50].

ELKI has a collection of data sets for outlier detection and also many data sets for OD methods evaluation. These data sets are used to study the performance of several OD algorithms and parameters.

4) *Unsupervised Anomaly Detection Dataverse* [49].

These datasets are used for evaluating unsupervised outlier detection algorithms by making comparison with the standards. It is obtained from multiple sources with the majority of the data sets from supervised machine learning datasets.

It is important to note that with real-world data sets, a lot of data is not publicly accessible due to privacy and security concerns.

Synthetic datasets are often created under the settings of defined constraints and conditions. Synthetic datasets, when compared to real-world datasets, are mostly less complex and eccentric, and shows better validity of the OD algorithms performance. For the outlier detection process, since most of the data adopted are not purpose-specific for just OD methods, the repurposing of supervised classification data has been widely adopted. In many studies, the data has been treated as it is, rather than manipulated.

As stated earlier, in outlier detection experiments, to evaluate the OD methods there is need to use both real-world and synthetic data sets. Also, many benchmark datasets are required to develop an algorithm that captures a broader view of the problems. The availability of many benchmark datasets also helps in the better and more robust way of reporting and presenting the results. In most supervised classification types of datasets, they require some preprocessing for outlier detection tasks. Two important aspects are considered in the preprocessing phase [226]. That is, for semantically significant outlier datasets, the outliers are the classes related to the minor objects and the normal data is the rest of the data.

When choosing a data set for OD methods, the data should be tailored in terms of precise and meaningful attributes which can fit the problem definition. For example, for an OD method related to the data stream, it is better to use

streaming data rather than other kinds of data. The selected algorithm should fit the data in terms of the right attribute types, the correct distribution model, the speed and scalability and other important anticipated incremental capabilities that can be managed and model well upon the arrival of new objects.

Some of the other concerns in dealing with datasets include how to handle the downsampling of data, dealing with duplicate data, transforming categorical attributes to numeric types, normalization, and dealing with missing values. In future work, it will be crucial to study how to evaluate dataset for outlier detection methods and what key attributes to take into consideration.

V. CONCLUSION AND OPEN RESEARCH GAPS

In this paper, we have provided a comprehensive survey in a structured manner that reviews state-of-the-art methods of detecting outliers by grouping them into different categories. We have grouped the algorithms into density-based, statistical-based, distance-based, clustering-based, ensemble-based, and learning-based approaches. In our discussion section, we discussed their most significant advantages, drawbacks, and challenges. Furthermore, we attempted to review and provide state-of-the-art open research problems and challenges. We also discussed the evaluation techniques, tools, and data sets adopted for outlier detection methods. We succeeded in providing researchers with an in-depth knowledge of the fundamental requirements of these techniques before choosing a particular technique for an outlier detection problem.

From our review, it is evident that despite the progress in outlier detection research, there are still lots of open research questions and issues to be addressed. It is apparent that future studies are needed in most of the outlier detection-based approaches. Therefore, in addition to the already stated future work in each of the categories, the following are still supplementary to open research gaps:

- Further studies need to be done to fully characterize and relate some of these methods to real-life data, particularly in very large and high dimensional databases, where first-hand techniques for estimating data densities are worth bearing in mind. In high dimensional data sets, the problem of the curse of dimensionality and distance concentration are still open challenges to be addressed.
- Outliers usually show unusual local behaviors. The process of discovering in high dimensional space these local correlations is challenging. Also, how to accurately determine the correlations makes the whole issue more complicated. Therefore, solving these challenges are still open research problems
- It would also be thought-provoking to examine the influence of extraneous features in outlier detection processes to select the appropriate features for the outlier detection task.
- Since a vast amount of data now comes in the form of data streams, which are characterized by some issues as mentioned earlier, it will be of interest for further research work to address these challenging issues to detect outliers more efficiently. In the existence of high dimensional data, most existing data stream algorithms for OD methods lose their effectiveness. Therefore, future studies are needed on how to redesign the contemporary models to detect the outlying patterns correctly and efficiently.
- The recent explosion of massive datasets, gives rise to many openings for future research relating to the design of efficient approaches to identify outliers, which are usually the most significant points within the data set. Their discoveries can lead to vital and unforeseen insights. We will also suggest the need for designing robust outlier detection algorithms that are scalable, can handle large dimensional data sets, and have a minimum run time.
- We found that in distance-based methods based on KNN, the parameter K is sensitive, therefore setting of the parameter k is very significant. Setting and finding the appropriate k is worth considering for neighbor ranking-based OD methods. Also, the distance metrics usually adopted for neighbor-based approaches do not fit well for high dimensional data. Addressing the equidistance issue and introducing effective distance metrics is necessary for high-dimensional data. The neighbor -based OD algorithms are sensitive to the nearest neighbors which are chosen for the models. Therefore, further studies can be done on how to determine the precise number of neighbors needed.
- In statistical-methods, apart from designing more robust algorithms for detecting outliers more efficiently, we noticed that, to the best of our knowledge, no work has been done to compare the influence of parametric and non-parametric approaches in the outlier detection process. It is essential for researchers to know the pros and cons of using the parametric and non-parametric approaches and also to design algorithms that can be able to outperform and address some of the drawbacks of statistical OD methods.
- For clustering techniques, since they are generally not considered to be designed explicitly for outlier detection, ensemble techniques, which combine the results from dissimilar models to produce a more robust model, will create a much better result. Ensemble methods are well known to improve the performance of outlier detection by both the quality of the detected outliers and run time. Therefore, ensemble outlier detection, which shows great potential in enhancing outlier detection algorithms, can be another worthy future research direction. More accurate models can be proposed to address the unexplored areas.
- For the learning methods such as subspace-based and ensemble-based learning methods, with a large range of the subspaces or base learners, we often get a reasonably good performance. Therefore, finding ways to choose

the precise subspaces and base learners is important. Also, choosing the right quantities and combination strategies are all still open research issues to address.

- In terms of evaluating OD methods, it is still an open challenge on how to effectively and broadly assess the OD methods performance. This has been difficult to achieve because outliers are not frequent and often the ground truth in real situations is absent.
- Another exciting open research direction for the future is the arrival or loss of new or existing dimensions over the period. In potential application areas such as outlier detection in IoT (internet of things) devices, the sensors can be on or off sporadically over the period, and new ways are needed to detect outliers more efficiently in this challenging scenario.
- Although considerable advances have been made in using deep learning methods in other application areas, However, there is a relative shortage of deep learning methods for outlier detection problems. Therefore, the use of deep learning techniques for OD methods is still open for further research.

REFERENCES

[1] E. L. Paula, M. Ladeira, R. N. Carvalho, and T. Marzagáo, "Deep learning anomaly detection as support fraud investigation in Brazilian exports and anti-money laundering" in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Anaheim, CA, USA, Oct. 2016, pp. 954–960.

[2] U. Porwal and S. Mukund, "Credit card fraud detection in e-commerce: An outlier detection approach," 2018, *arXiv:1811.02196*. [Online]. Available: https://arxiv.org/abs/1811.02196

[3] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Anaheim, CA, USA, Dec. 2016, pp. 195–200.

[4] G. Gebremeskel, C. Yi, Z. He, and D. Haile, "Combined data mining techniques based patient data outlier detection for healthcare safety," *Int. J. Intell. Comput. Cybern.*, vol. 9, no. 1, pp. 42–68, 2016.

[5] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.

[6] C. C. Aggarwal and P. S. Yu, "An effective and efficient algorithm for high-dimensional outlier detection," *Int. J. Very Large Data Bases*, vol. 14, no. 2, pp. 211–221, 2005.

[7] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 2, pp. 145–160, Feb. 2006.

[8] M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[9] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proc. SIAM Conf. Data Mining*, Apr. 2006, pp. 328–339.

[10] Z. Zheng, H. Y. Jeong, T. Huang, and J. Shu, "KDE based outlier detection on distributed data streams in multimedia network," *Multimedia Tools Appl.*, vol. 76, no. 17, pp. 18027–18045, Sep. 2017.

[11] L. L. Sheng, "Fractal-based outlier detection algorithm over RFID data streams," *Int. J. Online Eng.*, vol. 12, no. 1, pp. 35–41, Feb. 2016.

[12] D. van Hieu and P. Meesad, "A fast outlier detection algorithm for big datasets," in *Recent Advances in Information and Communication Technology* (Advances in Intelligent Systems and Computing), vol. 463, P. Meesad, S. Boonkrong, and H. Unger, Eds. Cham, Switzerland: Springer, 2016.

[13] X. T. Wang, D. R. Shen, M. Bai, T. Z. Nie, Y. Kou, and G. Yu, "An efficient algorithm for distributed outlier detection in large multi-dimensional datasets," *J. Comput. Sci. Technol.*, vol. 30, no. 6, pp. 1233–1248, Nov. 2015.

[14] A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, "Outlier detection approaches for wireless sensor networks: A survey," *Comput. Netw.*, vol. 129, pp. 319–333, Dec. 2017.

[15] J. Mao, W. Tao, C. Jin, and A. Zhou, "Feature grouping-based outlier detection upon streaming trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2696–2709, Dec. 2017.

[16] C. D'Urso, "EXPERIENCE: Glitches in databases, how to ensure data quality by outlier detection techniques," *J. Data Inf. Qual.*, vol. 7, no. 3, 2016, Art. no. 14.

[17] H. P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, Berlin, Germany: Springer, 2009, pp. 831–838.

[18] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 4, pp. 577–590, Aug. 2018.

[19] Y. Yu, L. Cao, E. A. Rundensteiner, and Q. Wang, "Outlier Detection over Massive-Scale Trajectory Streams," *ACM Trans. Database Syst.*, vol. 42, no. 2, pp. 10:1–10:33, 2017.

[20] Y. Djenouri, A. Belhadi, J. C.-W. Lin, D. Djenouri, and A. Cano, "A survey on urban traffic anomalies detection algorithms," *IEEE Access*, vol. 7, pp. 12192–12205, 2019.

[21] Y. Djenouri, A. Zimek, and M. Chiarandini, "Outlier detection in urban traffic flow distributions," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 935–940.

[22] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[23] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey," *Wiley Interdiscipl. Rev., Comput. Stat.*, vol. 7, no. 3, pp. 223–247, 2015.

[24] X. Su and T. C. Leroy, "Outlier detection," in *Robust Regression and Outlier Detection*, vol. 1, no. 3. Hoboken, NJ, USA: Wiley, 2011, pp. 261–268.

[25] J. Tamboli and M. Shukla, "A survey of outlier detection algorithms for data streams," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop.*, Mar. 2016, pp. 3535–3540.

[26] J. Zhang, "Advancement of outlier detection: A survey," *ICST Trans. Scalable Inf. Syst.*, vol. 13, pp. 1–26, Feb. 2013.

[27] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Stat. Anal. Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012.

[28] C. C. Aggarwal, *Outlier Analysis*. 2nd Ed. New York, NY, USA: Springer, 2016.

[29] A. S. Hadi, R. Imon, and M. Werner, *Detection of Outliers*, vol. 1, no. 1. Hoboken, NJ, USA: Wiley, 2009, pp. 57–70.

[30] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 10, pp. 1–13, Sep. 2017.

[31] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.

[32] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: https://arxiv.org/abs/1901.03407

[33] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *Comput. J.*, vol. 54, no. 4, pp. 570–588, Apr. 2011.

[34] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[35] L. F. Maimó, A. L. P. Gómez, F. G. J. Clemente, M. G. Pérez, and G. M. A. Pérez, "Self-adaptive deep learning-based system for anomaly detection in 5G networks," *IEEE Access*, vol. 6, pp. 7700–7712, 2018.

[36] I. Kakanakova and S. Stoyanov, "Outlier Detection via Deep Learning Architecture," *Proc. 18th Int. Conf. Comput. Syst. technol.*, vol. 2017, pp. 73–79.

[37] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2005, pp. 157–166.

[38] H. V. Nguyen, H. H. Ang, and V. Gopalkrishnan, "Mining outliers with ensemble of heterogeneous detectors on random subspaces," in *Database Systems for Advanced Applications*, Berlin, Germany: Springer, 2010, pp. 368–383.

[39] V. Barnett and T. Lewis, *Outliers in Statistical Data*. Hoboken, NJ, USA: Wiley, 1994.
[40] S. Walfish, "A review of statistical outlier methods," *Pharmaceutical Technol.*, vol. 30, no. 11, pp. 1–5, 2006.
[41] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.
[42] M. F. Jiang, S. S. Tseng, and C. M. Su, "Two-phase clustering process for outliers detection," *Pattern Recognit. Lett.*, vol. 22, pp. 691–700, May 2011.
[43] E. Achtert, H. P. Kriegel, L. Reichert, E. Schubert, R. Wojdanowski, and A. Zimek, "Visual evaluation of outlier detection models," in *Proc. 15th Int. Conf. Database Syst. Adv. Appl. (DASFAA)*, 2010, pp. 396–399.
[44] Alberto Quesada, Artelnics. *Three methods to deal with outliers, Artelnics, Machine Learning Blog*. Accessed: Feb. 20, 2018. [Online]. Available: https://www.neuraldesigner.com/blog/3_methods_to_deal_with_outliers
[45] K. Chenaoua, F. Kurugollu, and A. Bouridane, "Data cleaning and outlier removal: Application in human skin detection," in *Proc. 5th Eur. Workshop Vis. Inf. Process. (EUVIP)*, Dec. 2014, pp. 1–6.
[46] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Jul. 2018, pp. 2041–2050.
[47] H. Liu, X. Li, J. Li, and S. Zhang, "Efficient outlier detection for high-dimensional data," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 12, pp. 2451–2461, Dec. 2018.
[48] A. Emmott, S. Das, T. Dietterich, A. Fern, and W. K. Wong, "Anomaly detection meta-analysis benchmarks," in *Oregon State University*, San Francisco, CA, USA: Dataset, 2016. doi: 10.7267/N97H1GGX.
[49] M. Goldstein, *Unsupervised Anomaly Detection Benchmark*. Harvard Dataverse, 2015. doi: 10.7910/DVN/OPQMVF.
[50] (2018). *ELKI Outlier Datasets*. [Online]. Available: https://elki-project.github.io/datasets/outlier
[51] S. Rayana, (2018). *ODDS Library. Stony Brook, NY: Stony Brook University, Department of Computer Science*. [Online]. Available: http://odds.cs.stonybrook.edu
[52] K. Bache, M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[53] A. V. Pawar, P. P. Kalavadekar, and S. N. Tambe, "A survey on outlier detection techniques for credit card fraud detection," *IOSR J. Comput. Eng.*, vol. 16, no. 2, pp. 44–48, 2014.
[54] D. Huang, D. Mu, L. Yang, and X. Cai, "CoDetect: Financial fraud detection with anomaly feature detection," *IEEE Access*, vol. 6, pp. 19161–19174, 2018.
[55] G. Singh, F. Masseglia, C. Fiot, A. Marascu, and P. Poncelet, "Mining common outliers for intrusion detection," in *Advances in Knowledge Discovery and Management*, F. Guillet, G. Ritschard, D. A. Zighed, and H. Briand. (eds). Berlin, Germany: Springer, 2009, pp. 217–234.
[56] S. Cateni, V. Colla, and M. Vannucci, "Outlier detection methods for industrial applications," in *Advances in Robotics, Automation and Control*, J. Aramburo and A. R. Trevino Eds. Rijeka, Croatia: InTech, 2008. [Online]. Available: http://www.intechopen.com/books/advances_in_robotics_automation_and_control/outlier_detection_methods_for_industrial_applications
[57] A. Zhang, S. Song, J. Wang, and P. S. Yu, "Time series data cleaning: From anomaly detection to anomaly repairing," *Proc. VLDB Endowment*, vol. 10, no. 10, pp. 1046–1057, Jun. 2017.
[58] S. E. Benkabou, K. Benabdeslem, and B. Canitia, "Unsupervised outlier detection for time series by entropy and dynamic time warping," *Knowl. Inf. Syst.*, vol. 54, no. 2, pp. 463–486, Feb. 2018.
[59] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Effective and efficient trajectory outlier detection based on time-dependent popular route," *World Wide Web*, vol. 20, no. 1, pp. 111–134, Jan. 2017.
[60] J. Ramakrishnan, E. Shaabani, C. Li, and M. A. Sustik, "Anomaly detection for an e-commerce pricing system," 2019, *arXiv:1902.09566*. [Online]. Available: https://arxiv.org/abs/1902.09566
[61] R. Kannan, H. Woo, C. Charu Aggarwal, and H. Park, "Outlier detection for text data : An extended version," 2017, *arXiv:1701.01325*. [Online]. Available: https://arxiv.org/abs/1701.01325
[62] Y. Weng, N. Zhang, and C. Xia, "Multi-agent-based unsupervised detection of energy consumption anomalies on smart campus," *IEEE Access*, vol. 7, pp. 2169–2178, 2019.
[63] J. Lei, T. Jiang, K. Wu, H. Du, and L. Zhu, "Robust local outlier detection with statistical parameters for big data," *Comput. Syst. Sci. Eng.*, vol. 30, no. 5, pp. 411–419, 2015.
[64] R. Yu, H. Qiu, Z. Wen, C. Lin, and Y. Liu, "A survey on social media anomaly detection," *ACM SIGKDD Explor. Newslett.*, vol. 18, no. 1, pp. 1–14, 2016.
[65] R. H. X. Yu and Y. Liu, "Glad: Group anomaly detection in social media analysis," *ACM Trans. Knowl. Discovery Data (TKDD)*, vol. 10, no. 2, p. 18, 2015.
[66] S. Ghanbari, B. Ali Hashemi, and C. Amza, "Stage-aware anomaly detection through tracking log points," in *Proc. 15th Int. Middleware Conf.*, Dec. 2014, pp. 253–264. doi: 10.1145/2663165.2663319.
[67] K. Pradnya and H. K. Khanuja, "A methodology for outlier detection in audit logs for financial transactions," in *Proc. Int. Conf. Comput. Commun. Control Automat.*, Feb. 2015, pp. 837–840. doi: 10.1109/ICCUBEA.2015.167.
[68] A. Abid, A. Kachouri, and A. Mahfoudhi, "Outlier detection for wireless sensor networks using density-based clustering approach," *IET Wireless Sensor Syst.*, vol. 7, no. 4, pp. 83–90, Aug. 2017.
[69] H. Feng, L. Liang, and H. Lei, "Distributed outlier detection algorithm based on credibility feedback in wireless sensor networks," *IET Commun.*, vol. 11, no. 8, pp. 1291–1296, Jun. 2017.
[70] N. Shahid, I. H. Naqvi, and S. B. Qaisar, "Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: A survey," *Artif. Intell. Rev.*, vol. 43, no. 2, pp. 193–228, Feb. 2015.
[71] H. Zhang, J. Liu, and C. Zhao, "Distance based method for outlier detection in body sensor networks," *EAI Endorsed Trans. Wireless Spectr.*, vol. 2, no. 7, p. e4, Apr. 2016.
[72] M. Shukla, Y. P. Kosta, and P. Chauhan, "Analysis and evaluation of outlier detection algorithms in data streams," in *Proc. IEEE Int. Conf. Comput., Commun. Control (IC4)*, Sep. 2015, pp. 1–8.
[73] L. Tran, L. Fan, and C. Shahabi, "Distance-based outlier detection in data streams," in *Proc. VLDB Endowment (PVLDB)*, vol. 9, no. 12, pp. 1089–1100, Aug. 2016.
[74] E. Manzoor, H. Lamba, and L. Akoglu, "Outlier detection in feature-evolving data streams," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. Aug. 2018, pp. 1963–1972.
[75] W. Jin, A. K. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in *Proc. 10th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*, 2006, pp. 577–593.
[76] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2009, pp. 813–822.
[77] M. Bai, X. Wang, J. Xin, and G. Wang, "An Efficient algorithm for distributed density-based outlier detection on big data," *Neurocomputing*, vol. 181, pp. 19–28, Mar. 2016.
[78] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217303302
[79] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 190–237, 2014.
[80] J. Tang, Z. Chen, A. Fu, and D. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Advances in Knowledge Discovery and Data Mining*. Berlin, Germany: Springer, 2002, pp. 535–548.
[81] H. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local outlier probabilities," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, Nov. 2009, pp. 1649–1652.
[82] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proc. 19th Int. Conf. Data Eng.*, Mar. 2003, pp. 315–326.
[83] D. Ren, B. Wang, and W. Perrizo, "RDF: A density-based outlier detection method using vertical data representation," in *Proc. Int. Conf. Data Mining*, Nov. 2004, pp. 503–506.
[84] K. Cao, L. Shi, G. Wang, D. Han, and M. Bai, "Density-based local outlier detection on uncertain data," in *Web-Age Information Management. WAIM* (Lecture Notes in Computer Science), vol. 8485, F. Li, G. Li, S. Hwang, B. Yao, and Z. Zhang, Eds. Cham, Switzerland: Springer, 2014.
[85] F. Keller, E. Müller, and K. Bohm, "HiCS: High contrast subspaces for density-based outlier ranking," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, Apr. 2012, pp. 1037–1048.

[86] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans. Knowl. Discovery Data (TKDD)*, vol. 10, no. 1, 2015, Art. no. 5. doi: 10.1145/2733381.

[87] R. Momtaz, N. Mohssen, and M. A. Gowayyed, "DWOF: A robust density-based outlier detection approach," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, 2013, pp. 517–525.

[88] H. Fan, O. R. Zaïane, A. Foss, and J. Wu, "Resolution-based outlier factor: Detecting the top-n most outlying data points in engineering data," *Knowl. Inf. Syst.*, vol. 19, no. 1, pp. 31–51, 2009.

[89] K. Wu, K. Zhang, W. Fan, A. Edwards, and P. S. Yu, "RS-forest: A rapid density estimator for streaming anomaly detection," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 600–609.

[90] E. Lozano and E. Acufia, "Parallel algorithms for distance-based and density-based outliers," in *Proc. 5th IEEE Int. Conf. Data Mining*, Nov. 2005, pp. 729–732.

[91] F. I. Vázquez, T. Zseby, and A. Zimek, "Outlier detection based on low density models," *Proc. ICDM Workshops*, 2018, pp. 970–979.

[92] J. Ning, L. Chen, and J. Chen, "Relative density-based outlier detection algorithm," in *Proc. CSAI/ICIMT*, Dec. 2018, pp. 227–231.

[93] S. Su, L. Xiao, L. Ruan, F. Gu, S. Li, Z. Wang, and R. Xu, "An efficient density-based local outlier detection approach for scattered data," *IEEE Access*, vol. 7, pp. 1006–1020, 2019.

[94] W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc. 23rd VLDB Conf.*, Aug. 1997, pp. 186–195.

[95] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowl. Inf. Syst.*, vol. 26, no. 2, pp. 309–336, 2011.

[96] H. Kriegel, P. Kröger, and A. Zimek, "Outlier detection techniques," in *Proc. Tutorial KDD*, 2009, pp. 1–10.

[97] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS One*, vol. 11, no. 4, 2016, Art. no. e0152173. doi: 10.1371/journal.pone.0152173.

[98] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*. Jul. 2000, pp. 255–262.

[99] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *VLDB J.*, vol. 8, nos. 3–4, pp. 237–253, 2000.

[100] W. Contributors. (2015). *Maximum Likelihood Estimation [Internet]*. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Maximum_likelihood_estimation&oldid=857905834.

[101] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based GMM," in *Proc. SIAM Int. Conf. on Mining (SDM)*, Apr. 2009, pp. 145–154.

[102] X. Tang, R. Yuan, and J. Chen, "Outlier detection in energy disaggregation using subspace learning and Gaussian mixture model," *Int. J. Control Autom.*, vol. 8, no. 8, pp. 161–170, 2015.

[103] B. N. Saha, N. Ray, and H. Zhang, "Snake validation: A PCA-based outlier detection method," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 549–552, Jun. 2009.

[104] M. H. Satman, "A new algorithm for detecting outliers in linear regression," *Int. J. Statist. Probab.*, vol. 2, no. 3, pp. 101–109, Aug. 2013.

[105] C. M. Park and J. Jeon, "Regression-based outlier detection of sensor measurements using independent variable synthesis," in *Proc. Int. Conf. Data Sci.*, Dec. 2015, pp. 78–86.

[106] P. I. F. Dalatu, A. Fitrianto, and A. Mustapha, "A comparative study of linear and nonlinear regression models for outlier detection," in *Proc. Int. Conf. Soft Comput. Data Mining*, 2017, vol. 549, pp. 316–327.

[107] M. Pavlidou and G. Zioutas, "Kernel density outlier detector," in *Topics Nonparametric Statistics*. New York, NY, USA: Springer, 2014, pp. 241–250.

[108] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *Proc. 5th Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, 2007, pp. 61–75.

[109] J. Gao, W. Hu, Z. Zhang, X. Zhang, and O. Wu, "RKOF: Robust kernel-based local outlier detection," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2011, pp. 270–283.

[110] V. S. K. Samparthi and H. K. Verma, "Outlier detection of data in wireless sensor networks using kernel density estimation," *Int. J. Comput. Appl.*, vol. 5, no. 7, pp. 28–32, Aug. 2010.

[111] A. O. Boedihardjo, C.-T. Lu, and F. Chen, "Fast adaptive kernel density estimator for data streams," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 285–317, Feb. 2015.

[112] M. S. Uddin, A. Kuh, and Y. Weng, "Online bad data detection using kernel density estimation," in *Proc. IEEE Power Energy Sociaty General Meeting*, Jul. 2015, pp. 1–5.

[113] S. Smrithy, S. Munirathinam, and R. Balakrishnan, "Online anomaly detection using non-parametric technique for big data streams in cloud collaborative environment," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1950–1955.

[114] L. Zhang, J. Lin, and R. Karim, "Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowl.-Based Syst.*, vol. 139, pp. 50–63, Jan. 2018.

[115] X. Qin, L. Cao, E. A. Rundensteiner, and S. Madden, "Scalable kernel density estimation-based local outlier detection over large data streams," in *Proc. EDBT*, 2019, pp. 421–432. doi: 10.5441/002/edbt.2019.37.

[116] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," in *Proc. Poster Demo Track*, Sep. 2012, pp. 59–63.

[117] P. J. Rousseeuw and M. Hubert, "Robust statistics for outlier detection," *Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 73–79, 2011.

[118] H. Du, S. Zhao, and D. Zhang, "Robust local outlier detection," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 116–123.

[119] J. Gebhardt, M. Goldstein, F. Shafait, and A. Dengel, "Document authentication using printing technique features and unsupervised anomaly detection," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 479–483.

[120] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*, Sep. 2002, pp. 15–26.

[121] T. T. Dang, H. Y. T. Ngan, and W. Liu, "Distance-based k-nearest neighbors outlier detection method in large-scale traffic data," in *Proc. IEEE Int. Conf. Digital Signal Process.*, Jul. 2015, pp. 507–510.

[122] E. M. Knorr and R. T. Ng, "Algorithms for mining distance based outliers in large data sets," in *Proc. 24th Int. Conf. Very Large Databases Conf.*, 1998, pp. 392–403.

[123] S. Ramaswamy, R. Rastogi, and S. Kyuseok, "Efficient algorithms for mining outliers from large data sets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May 2000, pp. 427–438.

[124] A. Ghoting, S. Parthasarathy, and M. E. Otey, "Fast mining of distance-based outliers in high-dimensional datasets," *Data Mining Knowl. Discovery*, vol. 16, vol. 3, pp. 349–364, Jun. 2008.

[125] J. Liu and H. Deng, "Outlier detection on uncertain data based on local information," *Knowl. Based Syst.*, vol. 51, pp. 60–71, Oct. 2013.

[126] H. Huang, K. Mehrotra, and C. K. Mohan, "Rank-based outlier detection," *J. Stat. Comput. Simul.*, vol. 83, no. 3, pp. 518–531, Oct. 2013.

[127] G. Bhattacharya, K. Ghosh, and A. S. Chowdhury, "Outlier detection using neighborhood rank difference," *Pattern Recognit. Lett.*, vol. 60, pp. 24–31, Aug. 2015.

[128] X. Wang, X. L. Wang, Y. Ma, and D. M. Wilkes, "A fast MST-inspired *k*NN-based outlier detection method," *Inf. Syst.*, vol. 48, pp. 89–112, Mar. 2015.

[129] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1369–1382, May 2015.

[130] J. Huang, Q. Zhu, L. Yang, and J. Feng, "A non-parameter outlier detection algorithm based on natural neighbor," *Knowl. Based Syst.*, vol. 92, pp. 71–77, Jan. 2016.

[131] J. Ha, S. Seok, and J.-S. Lee, "A precise ranking method for outlier detection," *Inf. Sci.*, vol. 324, pp. 88–107, Dec. 2015.

[132] S. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2003, pp. 29–38.

[133] F. Angiulli and F. Fassetti, "Very efficient mining of distance-based outliers," *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, Nov. 2007, pp. 791–800.

[134] D. Ren, I. Rahal, W. Perrizo, and K. Scott, "A vertical distance-based outlier detection method with local pruning," in *Proc. 13th ACM CIKM Int. Conf. Inf. Knowl. Manage.*, Nov. 2004, pp. 279–284.

[135] N. H. Vu and V. Gopalkrishnan, "Efficient pruning schemes for distance-based outlier detection," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2009, pp. 160–175.

[136] F. Angiulli and F. Fassetti, "Distance-based outlier queries in data streams: The novel task and algorithms," *Data Mining Knowl. Discovery*, vol. 20, pp. 290–324, Mar. 2010.

[137] C. C. Aggarwal, "On abnormality detection in spurious populated data streams," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2005, pp. 80–91.
[138] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *Proc. Int. Conf. Very Large Data Bases*, Sep. 2006, pp. 187–198.
[139] D. Yang, E. A. Rundensteiner, and M. Ward, "Neighbor-based pattern detection for windows over streaming data," in *Proc. 12th Int. Conf. Extending Database Technol.*, Mar. 2009, pp. 529–540.
[140] M. Kontaki, A. Gounaris, A. N. Papadopoulos, and K. Tsichlas, "Continuous monitoring of distance-based outliers over data streams," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Apr. 2011, pp. 135–146.
[141] F. Angiulli and F. Fassetti, "Detecting distance-based outliers in streams of data," in *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, Nov. 2007, pp. 811–820.
[142] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner, "Scalable distance-based outlier detection over high-volume data streams," in *Proc. IEEE 30th Int. Conf. Data Eng.*, Apr. 2014, pp. 76–87.
[143] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis Tool," *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, Oct. 2011. [Online]. Available: https://moa.cms.waikato.ac.nz/
[144] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *ACM SIGMOD Rec.*, vol. 30, no. 2, pp. 37–46, 2001.
[145] K. Bhaduri, B. L. Mathews, and C. R. Giannella, "Algorithms for speeding up distance-based outlier detection," in *Proc. ACM KDD Conf.*, Aug. 2011, pp. 859–867.
[146] M. B. Al-Zoubi, "An effective clustering-based approach for outlier detection," *Eur. J. Sci. Res.*, vol. 28, no. 2, pp. 310–316, Jan. 2009.
[147] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 1990.
[148] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proc. 20th VLDB Conf.*, 1994, pp. 144–155.
[149] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist, Prob.*, Jun. 1967, vol. 1, pp. 281–297.
[150] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, no. 2, pp. 68–86, Jan. 1971.
[151] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 1998, pp. 73–84.
[152] G. Karypis, E. H. Han, and V. Kumar, "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling," *IEEE Comput.*, vol. 27, no. 3, pp. 329–341, Aug. 1999.
[153] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Aug. 1996, pp. 226–231.
[154] A. Hinneburg and D. A. Keim, "An efficient approach to cluster in large multimedia databases with noise," in *Proc. SIGKDD*, 1998, pp. 58–65.
[155] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A wavelet-based clustering approach for spatial data in very large databases," *VLDB J.*, vol. 8, nos. 3–4, pp. 289–304, Feb. 2000.
[156] J. Zhang, W. Hsu, and M. L. Lee, "Clustering in dynamic spatial databases," *J. Intell. Inf. Syst. (JIIS)*, vol. 24, no. 1, pp. 5–27, Jan. 2005.
[157] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1998, pp. 94–105.
[158] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," in *Proc. 30th Very Large Data Bases*, Aug. 2004, pp. 852–863.
[159] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in*Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2007, pp. 133–142.
[160] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proc. 29 th Int. Conf. Very Large Database*, vol. 29, pp. 81–92.
[161] J. Ren and R. Ma, "Density-based data streams clustering over sliding windows," in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Jul. 2009, pp. 248–252.
[162] I. Assent, P. Kranen, C. Baldauf, and T. Seidl, "AnyOut: Anytime outlier detection on streaming data," in *Proc. 17th Int. Conf. Database Syst. Adv. Appl.*, 2012, pp: 228-242.
[163] M. Elahi, K. Li, W. Nisar, X. Lv, and H. Wang, "Efficient clustering-based outlier detection algorithm for dynamic data stream," in *Proc. 5th Int. Conf. Fuzzy Syst. Knowl. Discovery*, vol. 5, Oct. 2008, pp. 298–304.
[164] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Apr. 2007, pp. 504–551.
[165] Yogita and D. Toshniwala, "A framework for outlier detection in evolving data streams by weighting attributes in clustering," in *Proc. 2nd Int. Conf. Commun., Comput. Secur.*, vol. 6, 2012, pp. 214–222. doi: 10.1016/j.protcy.2012.10.026.
[166] H. Moradi, S. Ibrahim, and J. Hosseinkhani, "Outlier detection in stream data by clustering method," *Int. J. Adv. Comput. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 25–34, 2013.
[167] S. V. Bhosale, "Outlier detection in streaming data using clustering approached," *Int. J. Adv. Comput. Sci. Inf. Technol.*, vol. 5, no. 5, pp. 6050–6053, 2014.
[168] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 1990.
[169] M. Moshtaghi, J. C. Bezdek, T. C. Havens, C. Leckie, S. S. Karunasekera, S. Rajasegarar, and M. Palaniswami, "Streaming analysis in wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 14, no. 9, pp. 905–921, Jun. 2014.
[170] M. Moshtaghi, J. C. Bezdek, C. Leckie, S. Karunasekera, and M. Palaniswami, "Evolving fuzzy rules for anomaly detection in data streams," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 3, pp. 688–700, 2015.
[171] M. Salehi, C. A. Leckie, M. Moshtaghi, and T. Vaithianathan, "A relevance weighted ensemble model for anomaly detection in switching data streams," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2014, pp. 461–473.
[172] M. Chenaghlou, M. Moshtaghi, C. Leckie, and M. Salehi, "An efficient method for anomaly detection in non-stationary data streams," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.
[173] H. Rizk, S. Elgokhy, and A. Sarhan, "A hybrid outlier detection algorithm based on partitioning clustering and density measures," in *Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2015, pp. 175–181.
[174] M. Chenaghlou, M. Moshtaghi, C. Leckie, and M. Salehi, "Online clustering for evolving data streams with online anomaly detection," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Springer, 2018, pp. 508–521.
[175] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *Proc. 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 625–636.
[176] A. Zimek, M. Gaudet, R. J. Campello, and J. Sander, "Subsampling for efficient and effective unsupervised outlier detection ensembles," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 428–436.
[177] J. R. Pasillas-Díaz and S. Ratté, "Bagged subspaces for unsupervised outlier detection," *Int. J. Comput. Intell.*, vol. 33, no. 3, pp. 507–523, Aug. 2017.
[178] C. C. Aggarwal, "Outlier ensembles: Position paper," *SIGKDD Explor. Newslett.*, vol. 14, pp. 49–58, Apr. 2013.
[179] A. Zimek, M. Gaudet, R. J. Compello, and J. Sander, "Subsampling for effcient and effective unsupervised outlier detection ensembles," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 428–436.
[180] A. Zimek, R. J. Campello, and J. Sander, "Data perturbation for outlier detection ensembles," in *Proc. 26th Int. Conf. Sci. Stat. Database Manage.*, Jul. 2014, pp. 1–13. doi: 10.1145/2618243.2618257.
[181] A. Zimek, R. J. G. B. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: Challenges and research questions a position paper," *ACM SIGKDD Explor. Newslett.*, vol. 15, no. 1, pp. 11–22, Jun. 2014.
[182] C. C. Aggarwal and S. Sathe, "Theoretical foundations and algorithms for outlier ensembles," *ACM SIGKDD Explor. Newslett.*, vol. 17, no. 1, pp. 24–47, Jun. 2015.
[183] Y. Zhao and M. K. Hryniewicki, "XGBOD: Improving supervised outlier detection with unsupervised representation learning," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2018, pp. 1–8. doi: 10.1109/IJCNN.2018.8489605.
[184] S. Rayana and L. Akoglu, "Less is more: Building selective anomaly ensembles," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 4, pp. 1–33, Jul. 2016.

[185] S. Rayana, W. Zhong, and L. Akoglu, "Sequential ensemble learning for outlier detection: A bias-variance perspective," in *Proc. ICDM*, Dec. 2017, pp. 1167–1172.

[186] B. M. B. Micenková and I. Assent, "Learning representations for outlier detection on a budget," 2015, *arXiv:1507.08104*. [Online]. Available: https://arxiv.org/abs/1507.08104

[187] G. O. Campos, A. Zimek, and W. Meira, Jr., "An unsupervised boosting strategy for outlier detection ensembles," in *Proc. 22nd Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining (PAKDD)*, 2018, pp. 564–576.

[188] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. 4th IEEE Int. Conf. Data Mining*, Apr. 2004, pp. 19–26.

[189] E. Müller, S. Gunnemann, T. Seidl, and I. Farber, "Discovering multiple clustering solutions: Grouping objects in different views of the data," in *Proc. 28th IEEE ICDE Conf.*, Apr. 2012, pp. 1207–1210.

[190] H. P. Kriegel, P. Kréger, E. Schubert, and A. Zimek, "Interpreting and unifying outlier scores," in *Proc. SDM*, Apr. 2011, pp. 13–24.

[191] E. Schubert, R. Wojdanowski, A. Zimek, and H. P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proc. SDM*, Apr. 2012, pp. 1047–1058.

[192] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Jul. 2008, pp. 413–42. doi: 10.1109/ICDM.2008.17.

[193] S. Das, W. K. Wong, T. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 853–858. doi: 10.1109/ICDM.2016.0102.

[194] T. Pimentel, M. Monteiro, and J. Viana, "A generalized active learning approach for unsupervised anomaly detection," vol. 2018. *arXiv:1805.09411*. [Online]. Available: https://arxiv.org/abs/1805.09411

[195] J. Zhang, Y. Jiang, K. H. Chang, S. Zhang, J. Cai, and L. Hu, "A concept lattice based outlier mining method in lowdimensional subspaces," *Pattern Recognit. Lett.*, vol. 30, no. 15, pp. 1434–1439, Nov. 2009.

[196] J. K. Dutta, B. Banerjee, and C. K. Reddy, "RODS: Rarity based outlier detection in a sparse coding framework," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 483–495, Feb. 2016.

[197] J. Zhang, S. Zhang, K. H. Chang, and X. Qin, "An outlier mining algorithm based on constrained concept lattice," *Int. J. Syst. Sci.*, vol. 45, no. 5, pp. 1170–1179, May 2014.

[198] E. Müller, I. Assent, U. Steinhausen, and T. Seidl, "OutRank: Ranking outliers in high dimensional data," in *Proc. IEEE 24th Int. Conf. Data Eng. Workshop*, Apr. 2008, pp. 600–603.

[199] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Trans. Knowl. Data Eng.*, to be published. doi: 10.1109/TKDE.2019.2905606.

[200] N. Gornitz, M. Kloft, K. Rieck, and U. Bredfeld, "Toward supervised anomaly detection," *J. Artif. Intell. Res.*, vol. 46, pp. 235–262, 2013.

[201] S. Das, M. R. Islam, N. K. Jayakodi, and J. R. Doppa, "Active anomaly detection via ensembles: Insights, algorithms, and interpretability," 2019. *arXiv:1901.08930*. [Online]. Available: https://arxiv.org/abs/1901.08930

[202] E. Müller, M. Schiffer, and T. Seidl, "Statistical selection of relevant subspace projections for outlier ranking," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Apr. 2011, pp. 434–445.

[203] B. V. Stein, M. van Leeuwen, and T. Bäck, "Local subspace-based outlier detection using global neighbourhoods," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1136–1142.

[204] H. D. K. Moonesinghe and P.-N. Tan, "Outrank: A graph-based outlier detection framework using random walk," *Int. J. Artif. Intell. Tools*, vol. 17, no. 1, pp. 19–36. doi: 10.1142/s0218213008003753.

[205] C. Wang, H. Gao, Z. Liu, and Y. Fu, "A new outlier detection model using random walk on local information graph," *IEEE Access*, vol. 6, pp. 75531–75544, 2018.

[206] W. Chao, G. Hui, L. Zhen, and F. Yan, "Outlier detection using diverse neighborhood graphs," in *Proc. 15th Int. Comput. Conf. Wavelet Act. Media Technol.*, 2018, pp. 58–62. doi: 10.1109/ICCWAMTIP.2018.8632604.

[207] B. R. Kiran, D. M. Thomas, and R. Parakkal, "An overview of deep learning-based methods for unsupervised and semi-supervised anomaly detection in videos," *J. Imag.*, vol. 4, no. 2, p. 36, 2018.

[208] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2018, pp. 387–395.

[209] D. Li, D Chen, L. Shi, B. Jin, J. Goh, and S. K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," 2019, *arXiv:1901.04997*. [Online]. Available: https://arxiv.org/abs/1901.04997

[210] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–19.

[211] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 665–674.

[212] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," 2017, *arXiv:1704.06743*. [Online]. Available: https://arxiv.org/abs/1704.06743

[213] J. T. A. Andrews, E. J. Morton, and L. D. Griffin, "Detecting anomalous data using auto-encoders," *Int. J. Mach. Learn. Comput.*, vol. 6, no. 1, pp. 21–26, 2016.

[214] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and largescale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.

[215] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," 2019, *arXiv:1812.04606*. [Online]. Available: https://arxiv.org/abs/1812.04606

[216] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1285–1298.

[217] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly Detection Using Autoencoders in high performance computing systems," 2018, *arXiv:1811.05269*. [Online]. Available: https://arxiv.org/abs/1811.05269

[218] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," 2018, *arXiv:1802.06360*. [Online]. Available: https://arxiv.org/abs/1802.06360

[219] L. Ruff, N. Gornitz, L. Deecke, S. A. Siddiqui, R. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 4390–4399.

[220] G. H. Orair, C. Teixeira, Y. Wang, W. Meira Jr, and S. Parthasarathy, "Distance-based outlier detection: Consolidation and renewed bearing," *VLDB Endowment*, vol. 3, no. 2, pp. 1469–1480, 2010.

[221] (2018). *ELKI Tutorials - The Demonstrated Software is Available As Release 0.3 of the ELKI Framework*. [Online]. Available: https://elki-project.github.io/tutorial/

[222] E. Achtert, T. Bernecker, H. P. Kriegel, E. Schubert, and A. Zimek, "ELKI in Time: ELKI 0.2 for the performance evaluation of distance measures for time series," in *Advances in Spatial and Temporal Databases*, N. Mamoulis, T. Seidl, T. B. Pedersen, K. Torp, and I. Assent, (eds.) Heidelberg, Germany: Springer, 2009, pp. 436–440.

[223] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognit.*, vol. 74, pp. 406–421, Feb. 2018.

[224] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313–327, Jul. 2014.

[225] U. Carrasquilla, "Benchmarking algorithms for detecting anomalies in large datasets," *CMG J.*, vol. 1, pp. 1–16, Nov. 2011.

[226] G. O. Campos, A. Zimek, and J. Sander, "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *J. Data Mining Knowl. Discovery*, vol. 30, no. 4, pp. 891–927, Jul. 2016.

[227] Y. Zhao and M. K. Hryniewicki, "Dcso: Dynamic combination of detector scores for outlier ensembles," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2018, pp. 1–8.

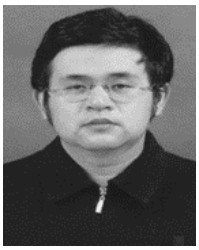
[228] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki, and Z. Li, "LSCP: Locally selective combination in parallel outlier ensembles," in *Proc. SIAM Int. Conf. Data Mining*, May 2019, pp. 585–593.

[229] C. C. Aggarwal, "Outlier ensembles," *ACM SIGKDD Explor. Newslett.*, vol. 14, no. 2, pp. 49–80, 2017.

[230] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms—The numenta anomaly benchmark," in *Proc. 14th Int. Conf. Mach. Learn. Appl.*, Dec. 2015, pp. 38–44.

[231] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[232] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," 2019, *arXiv:1901.01588*. [Online]. Available: https://arxiv.org/abs/1901.01588
[233] I. Kalayci and T. Ercan, "Anomaly detection in wireless sensor networks data by using histogram based outlier score method," in *Proc. 2nd Int. Symp. Multidisciplinary Studies Innov. Technol. (ISMSIT)*, Oct. 2018, pp. 1–6.
[234] M. Goldstein, M. Amer, J. Gebhardt, P. Kalka, and A. Elsawy. (2018). *RapidMiner Anomaly Detection Extension*. [Online]. Available: https://github.com/Markus-Go/rapidminer-anomaly detection
[235] *Detect and Remove Outliers in Data*. Accessed: Oct. 15, 2018. [Online]. Available: https://www.mathworks.com/help/matlab/ref/rmoutliers.html
[236] E. Kirner, E. Schubert, and A. Zimek, "Good and bad neighborhood approximations for outlier detection ensembles," in *Proc. Int. Conf. Similarity Search Appl.*, 2017, pp. 173–187.
[237] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newslett.*, vol. 19, no. 1, pp. 22–36, 2017.
[238] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proc. SIAM Int. Conf. Data Mining, Soc. Ind. Appl. Math.*, Jul. 2017, pp. 90–98.
[239] T. Xiao, C. Zhang, and H. Zha, "Learning to detect anomalies in surveillance video," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1477–1481, Sep. 2015.
[240] A. Koufakou and M. Georgiopoulos, "A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes," *Data Mining Knowl. Discovery*, vol. 20, no. 2, pp. 259–289, 2010.
[241] C. C. Aggarwal and P. S. Yu, "Outlier detection with uncertain data," in *Proc. SIAM Int. Conf.*, 2008, pp. 483–493.



HONGZHI WANG is currently a Professor and a Doctoral Supervisor with the Harbin Institute of Technology. He was awarded the Microsoft Fellowship, the Chinese Excellent Database Engineer, and the IBM Ph.D. Fellowship. He has published more than 200 papers in refereed journals and conferences. His research interests include big data management, data mining, data quality, and graph data management.

MOHAMED JAWARD BAH received the B.Eng. degree in electrical and electronics engineering from the University of Sierra Leone, in 2013, and the M.Sc. degree in computer science from the Nanjing University of Information Science and Technology, China, in 2016. He is currently pursuing the Ph.D. degree with the Harbin Institute of Technology, China. His research interests include data mining, and outlier detection in data streams and big data.

MOHAMED HAMMAD received the M.Sc. degree from the Information Technology Department, Faculty of Computers and Information, Menoufia University, Egypt, in 2015. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. He has been a Demonstrator and an Assistant Lecturer with the Faculty of Computers and Information, Menoufia University, since April 2012. His research interests include computer vision, machine learning, pattern recognition, and biometrics.