Hindawi
Complexity
Volume 2019, Article ID 2686378, 11 pages
https://doi.org/10.1155/2019/2686378

Review Article

Recent Progress of Anomaly Detection

Xiaodan Xu,[1,2] Huawen Liu,[1,3] and Minghai Yao [2]

[1]*Department of Computer Science, Zhejiang Normal University, Jinhua 321004, China*
[2]*College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310000, China*
[3]*Shanghai Key Laboratory of Intelligent Information Processing, Shanghai 200433, China*

Correspondence should be addressed to Minghai Yao; ymh@zjut.edu.cn

Received 10 October 2018; Revised 11 December 2018; Accepted 31 December 2018; Published 13 January 2019

Guest Editor: David Gil

Anomaly analysis is of great interest to diverse fields, including data mining and machine learning, and plays a critical role in a wide range of applications, such as medical health, credit card fraud, and intrusion detection. Recently, a significant number of anomaly detection methods with a variety of types have been witnessed. This paper intends to provide a comprehensive overview of the existing work on anomaly detection, especially for the data with high dimensionalities and mixed types, where identifying anomalous patterns or behaviours is a nontrivial work. Specifically, we first present recent advances in anomaly detection, discussing the pros and cons of the detection methods. Then we conduct extensive experiments on public datasets to evaluate several typical and popular anomaly detection methods. The purpose of this paper is to offer a better understanding of the state-of-the-art techniques of anomaly detection for practitioners. Finally, we conclude by providing some directions for future research.

1. Introduction

Anomaly analysis is of great interest to diverse research fields, including data mining and machine learning. It aims to identify those regions from data whose behaviours or patterns do not conform to expected values [1]. The unexpected behaviours, which are significantly different from those of the remainder of the given data, are commonly called anomalies. Notwithstanding, there is no widely acceptable formal definition of this concept. In the literature, an anomaly is also referred to as an outlier, a discordant object, an exception, an aberration, or a peculiarity, depending on specific application scenarios [1–5].

Identifying interesting or unexpected patterns is very important to many domains, such as decision making, business intelligence, and data mining. For example, an abnormal network transmission may imply that a computer system is attacked by hackers or viruses, an anomalous transaction of a credit card may imply unauthorized usage, and unexpected geological activity in nature can be a precursor of an earthquake or tsunami. Due to this fact, anomaly detection has a wide variety of applications, including public medical health, credit card fraud and network intrusion, and data cleaning [3, 5].

With the emergence of new technologies, data collected from real-world scenarios are becoming larger and larger, not only in size, but also in dimensionality. The high-dimensional property makes the data objects almost equidistant to each other. This implies that any data objects become very close as the dimensionality of data increases, resulting in the meaningless nature of their respective distances [4]. In this case, traditional anomaly detection methods cannot effectively handle high-dimensional data. In addition, most of the traditional detection methods assume that the data have the same type of features. However, the data in reality often have different feature types, such as numerical, binary, categorical, or nominal. This leads to an increased difficulty in anomaly detection.

Since anomaly detection has a wide range of potential applications, a great number of detection algorithms have been witnessed during the past decades. In this paper, we briefly review the latest works and place especial focuses on the ones for those complex data with high dimensionalities and mixed types. Generally, the existing anomaly detection

Table 1: A brief description of the anomaly detection methods.

Types	Descriptions & Typical methods	Advantages	Disadvantages
Neighbour-based detection	Identifying anomalies by using neighbourhood information. Typical examples include *k*NN[9], *k*NNW[10], LOF[11], LoOP[12], ODIN[13], RBDA[6], etc.	(i) Independent of the data distributions (ii) Intuitively understood and easily interpreted	(i) Sensitive to parameters (ii) Relatively poor performance
Subspace-based detection	Finding anomalies by sifting through different feature subsets. Representative examples include SOD[7], Zhang et al. [14, 15], RODS[16], OR[17], Muller et al. [18], etc.	(i) High efficiency (ii) Very effectiveness in some cases	(i) Finding the relevant feature subspaces for outliers is nontrivial and difficult
Ensemble-based detection	Integrating various anomaly detection results to achieve a consensus. Representatives are FB [19], HiCS [8], Stein et al. [20], Zimek et al. [21], Passillas et al. [22], and so on.	(i) High accuracy (ii) Less sensitive	(i) Inefficient (ii) Choosing the right meta-detectors is difficult
Mixed-type detection	Making a unified model for different data types, or taking each data type separately. Classical examples have LOADED [23], ODMAD [24], Zhang et al. [25], Lu et al. [26], Do et al. [27], and so on.	(i) Capable of handling the data with different types (ii) Relatively high accuracy	(i) Obtaining the correlation structures of features is difficult (ii) High complexity

techniques can be grouped into three categories: neighbour-based, subspace-based, and ensemble-based detection methods, depending on the techniques used. Table 1 summaries brief descriptions of the anomaly detection algorithms, including their definitions, pros, and cons.

In the literature, there are several survey papers (e.g., [1–5]) proposed for anomaly detection. However, they concern different aspects of anomaly detection. For example, [1] only reviews traditional outlier detection algorithms, while [2] places its focus on ensemble learning ones. The detection methods for specific application domains like network data and temporal data have been overviewed in [5] and [3], respectively. Unlike the surveys above, this paper only involves the latest and popular anomaly detection methods for the data with high dimensionality and mixed types, on which the classical detection methods cannot handle very well. Besides, this paper also offers more information related to anomaly detection, such as public datasets and widely used metrics. These aspects, however, have not been considered in the other papers. Additionally, this paper has made a comprehensively experimental comparison of several popular detection algorithms. The paper aims to help practitioners to better understand the state-of-the-art techniques of anomaly detection.

The remainder of this paper is organized as follows. Section 2 presents a survey of anomaly detection for the complicated data, including neighbour-based, subspace-based, and ensemble-based detection techniques. Section 3 provides evaluation metrics commonly used in the anomaly detection techniques, followed by experimental comparisons of the popular detection methods in Section 4. Section 5 concludes the paper.

2. Methodology

How to effectively identify outliers from the high-dimensional or mixed-type data is a fundamental and challenging issue in outlier detection. Recently, a rich number of detection algorithms have been developed to alleviate the problems. Roughly speaking, they can be divided into three categories, that is, neighbour-based (e.g., RBDA [6]), subspace-based (e.g., SOD [7]), and ensemble-based methods (e.g., HiCS [8]). The neighbour-based outlier detection methods mainly exploit the neighbourhood information of a given data object to determine whether it is far from its neighbours or its density is low or not. The subspace-based detection methods identify anomalies by sifting through different feature subsets in an ordered way. Unlike the routine algorithms, the ensemble-based ones combine the outputs of several detection algorithms or base detectors into a unified output by using integrated strategies. Table 1 briefly summarizes descriptions of the anomaly detection techniques.

2.1. Neighbour-Based Detection. The basic idea of the neighbour-based anomaly detection methods is to identify outliers by virtue of the neighbourhood information. Given a data object, the anomaly score is defined as the average distance (*k*NN [9]) or weighted distance (*k*NNW [10]) to its k nearest neighbours. Another strategy is to take the local outlier factor (LOF) [11] as the measurement of anomaly degree, in which the anomaly score was measured relative to its neighbourhoods. Based on LOF and LoOP [12] provided for each object an outlier probability as score, which is easily interpretable and can be compared over one data set. In

ODIN (Outlier Detection using Indegree Number) [13], an object is defined as an outlier if it participates in most neighbourhoods in kNN graph.

Note that all the neighbour-based detection methods mentioned above are independent of the distributions of the data and capable of detecting isolated objects. However, their performance heavily relies on the distance measures, which become unstable or meaningless in high-dimensional spaces. To cope with this problem, a feasible solution is to consider the ranking of neighbours, because, for each object, the ranking of its nearest neighbours is still meaningful to the nature of high-dimensional data. The underlying assumption is that two objects would most likely become nearest neighbours or have similar neighbours if they were generated from the same mechanism [7]. Following this idea, RBDA (Rank-Based Detection Algorithm) [6] takes the ranks of each object in its neighbours as the proximity degree of the object. For each object $s \in D$, let $N_k(s)$ be the k nearest neighbours of s. The anomaly degree of s is defined as follows:

$$A_k(s) = \frac{\sum_{p \in N_k(s)} r_p(s)}{\|N_k(s)\|} \quad (1)$$

where $r_p(s)$ is the rank of s among the neighbours of p. According to Eq. (1), one may observe that if s ranks behind the neighbours $N_k(s)$, it has a higher anomaly degree and would have a high probability of being considered an anomaly. RBDA does not consider the distance information of objects with regard to their neighbours, which would be useful in some cases; MRD (Modified-Ranks with Distance) [28] does. MRD takes both the ranks and the distances into account when estimating the anomaly scores of objects.

A special kind of the nearest neighbour, called the reverse neighbour, is also used to represent the proximate relationship among objects. For any object s, p is called a reverse neighbour of s if s is one of the nearest neighbours of p, and vice versa, that is, $s \in N_k(p)$ and $p \in N_k(s)$. The intuitive idea is that if an object has fewer reverse nearest neighbours, it is more likely to be an anomaly. Radovanovic et al. [29] adopted the reverse nearest neighbours to estimate the anomaly scores for each object. Bhattacharya et al. [30] continued this method even further by adopting both the reverse neighbours and the ranks of nearest neighbours to measure the anomaly score for each candidate object. Zhang et al. [31] estimated the anomaly scores using the number of the shared nearest neighbours of objects. Tang and He [32] exploited three kinds of neighbourhoods, including k nearest neighbours, reverse nearest neighbours, and shared nearest neighbours, to determine the anomaly scores in the local kernel density estimation. The neighbour ranking-based methods are sensitive to k, where different k values will yield different results. In addition, assigning the right value to k for a specific application is not trivial. To this end, Ha et al. [33] adopted a heuristic strategy to select an appropriate value for k using an iterative random sampling procedure. The assumption is that outlying objects are less likely to be selected than inlying objects in random sampling. Thus, greater inlier scores, called the observability factor (OF), should be given to the selected objects in each sampling. After several iterations of random sampling, the OF score of each object is estimated by counting its occurrence times in its neighbourhood. Based on the OF scores, the value of k can be appropriately assigned as the entropy of the observability factors.

2.2. Subspace-Based Detection. Anomalies often exhibit unusual behaviours in one or more local or low-dimensional subspaces. The low-dimensional or local abnormal behaviours would be masked by full dimensional analysis [34]. Zimek et al. [4] noted that, for an object in a high-dimensional space, only a subset of relevant features offers valuable information, while the rest are irrelevant to the task. The existence of the irrelevant features may impede the separability of the anomaly detection model. However, the anomaly detection techniques discussed so far identify anomalous objects from the whole data space with full dimensions. Thus, identifying anomalies from appropriate subspaces appears to be more interesting and efficient.

Subspace learning is a popular technique to handle high-dimensional problems in the literature. It is also extensively studied in anomaly analysis. The anomaly detection methods based on subspace techniques aim at finding anomalies by sifting through different subsets of dimensions in an ordered way. These methods have two kinds of representations: the sparse subspace methods [14, 16, 35, 36] and the relevant subspace methods [7, 15, 17, 18, 37].

The sparse subspace techniques project all objects in a high-dimensional space onto one or more low-dimensional and sparse subspaces. The objects falling into the sparse subspaces are considered anomalies because the sparse subspaces have abnormally lower densities. It is noted that exploring the sparse projections from the entire high-dimensional space is a time-consuming process. To alleviate this problem, Aggarwal and Yu [36] exploited an evolutionary algorithm to improve the exploration efficiency, where a subspace with the most negative scarcity coefficients was considered a space projection. However, the performance of the evolutionary algorithm heavily relies on some factors, such as the initial populations, the fitness functions, and selection methods.

Subspace representation and encoding are another studied topic for sparse subspace techniques. As a typical example, Zhang et al. [14] utilized the concept of lattice to represent the relationship of subspaces, where the subspaces with low density coefficients are regarded as sparse ones. This kind of method shows advantages in the performance and the completeness. However, constructing the concept lattice of subspaces is complex, leading to low efficiency. Dutta et al. [16] leveraged the technique of sparse encoding to project objects to a manifold space with a linear transformation, making the space sparse.

The relevant subspace methods exploit local information represented as relevant features to identify anomalies. For instance, OR (Out Ranking) [17] extend a subspace clustering model to rank outliers in heterogeneous high-dimensional data. SOD (Subspaces Anomaly Detection) [7] is a typical example of the relevant subspace learning methods. It first explores several correlation datasets by using the shared nearest neighbours for each object and then determines an

axis-parallel subspace on each correlation dataset by linear correlation such that each feature has low variance in the subspace. Unlike SOD, Muller et al. [37] used the relevant relationships of features from the correlation dataset to determine the subspace. Specifically, they obtained relevant subspaces by examining the relevant relationships of features with the Kolmogorov-Smirnov test [38]. Then, the anomaly degree of the object was calculated by multiplying the local anomaly scores in each relevant subspace. It can be easily observed that this kind of detection method is computationally expensive. The limitation of this method is that it requires a great number of local data to detect the trend of deviation.

2.3. Ensemble-Based Detection. Ensemble learning is widely studied in machine learning [39, 40]. Since it has a relatively better performance than other related techniques, ensemble learning is also frequently used for anomaly detection. As we know, none of the outlier detection methods can discover all anomalies in a low-dimensional subspace due to the complexity of the data. Thus, different learning techniques or even multiple subspaces are required simultaneously, where the potential anomalies are derived by ensemble techniques. In the literature, there are two ensemble strategies frequently adopted for anomaly analysis, that is, summarizing the anomaly scores and selecting the best one after ranking. For anomaly analysis, feature bagging and subsampling are extensively studied.

The FB (Feature Bagging) detection method [19] aims to train multiple models on different feature subsets sampled from a given high-dimensional space and then combines the model results into an overall decision. A typical example of this technique is the work done by Lazarevic and Kumar [19], in which feature subsets are randomly selected from the original feature space. On each feature subset, the score of each object is estimated with an anomaly detection algorithm. Then, the scores for the same object are integrated as the final score. Nguyen et al. [41] used different detection techniques, rather than the same one, to estimate anomaly scores for each object on random subspaces.

Keller et al. [8] proposed a flexible anomaly detection method that decouples the process of anomaly mining into two steps, that is, subspace search and anomaly ranking. The subspace search aims at obtaining high contrast subspaces (HiCS) using the Monte Carlo sampling technique, and, then, the LOF scores of objects are aggregated upon the obtained subspaces. Stein [20] extended this by first gathering the relevant subspaces of HiCS and then calculated the anomaly scores of objects using local anomaly probabilities (LoOP) [12], in which the neighbourhood is selected in the global data space.

The subsampling technique obtains training objects from a given collection of data without replacement. If implemented properly, it can effectively improve the performance of detection methods. For example, Zimek et al. [21] applied the technique of random subsampling to obtain the nearest neighbours for each object and then estimated its local density. This ensemble method, coupled with an anomaly detection algorithm, has a higher efficiency and provides a diverse set of results.

There are several anomaly detection methods that consider both feature bagging and subsampling. For example, Pasillas-Diaz et al. [22] obtained different features at each iteration via feature bagging and then calculated the anomaly scores for different subsets of data via subsampling. However, the variance of objects is difficult to obtain using feature bagging, and the final results tend to be sensitive to the size of subsampled datasets.

2.4. Mixed-Type Detection. It is worthy of remark that most of the anomaly detection methods mentioned above can only handle numerical data, resulting in poor robustness. In real-world applications, categorical and nominal features are ubiquitous; that is, categorical and numerical features are mixed within the same dataset [34]. Such mixed-type data pose great challenges to the existing detection algorithms. For mixed-type data, a common and simple strategy is to discretize numerical features and then treat them as categorical ones so that the detection methods for categorical data can be applied directly. While this practice seems to be a good solution, it may lose important information, that is, the correlations between features, leading to poor performance.

By now, a great number of detection methods have been developed to handle categorical data in the literature [42]. For example, He et al. [43] proposed a frequent pattern-based anomaly detection algorithm, where the potential anomalies were measured using a frequent pattern anomaly factor. As a result, the data objects that contained infrequent patterns could be considered anomalies. Contrastively, Otey et al. [44] developed a nonfrequent item set-based anomaly detection algorithm. Despite the pattern-based methods being suitable for handling categorical data, they are time consuming for general cases. Wu and Wang [45] estimated the frequent pattern anomaly factors based on nonexhaustive methods by mining a small number of patterns instead of all the frequent patterns. Koufakou and Georgiopoulos [46] considered the condensed representation of nonderivable item sets in their algorithm, which is a compact representation and can be obtained less expensively.

There are a lot of studies attempting to handle mixed-type data directly in the literature. Typical examples include LOADED [23], RELOADED, and ODMAD [24]. For instance, LOADED calculates an anomaly score for each object by using the support degrees of item sets for categorical features and correlation coefficients for numerical features [23]. RELOAD employs naive Bayes classifiers to predict abnormalities of categorical features. Finally, ODMAD treats categorical and numerical features separately. Specifically, it first calculates anomaly scores for categorical features using the same classification algorithm as LOADED. The objects, which are not identified as anomalies at this step, will be examined over numerical features with the cosine similarity [24]. Bouguessa [47] modelled the categorical and numerical feature space by using a mixture of bivariate beta distributions. The objects having a small probability of belonging to any components are regarded as anomalies.

The correlations of features have also been taken into consideration. For example, Zhang and Jin [25] exploited the concept of patterns to determine anomalies. In this method,

a pattern is a subspace formed by a particular category and all numerical features. Within this context, the patterns are learned via logistic regression. The objects would be considered anomalies if the probability returned by the model is far from a specific pattern. Lu et al. [26] took pairwise correlations of mixed-type features into consideration and presented a generalized linear model framework for anomaly analysis. Additionally, the t-student distribution was also used to capture variations of anomalies from normal objects. More recently, Do et al. [27] calculated anomaly scores for each object using the concept of free energy derived from a mixed-variant restricted Boltzmann machine. Since this well captured the correlation structures of mixed-type features through the factoring technique, it has a relatively high performance.

3. Evaluation Measurements

Unlike the problems of classification, evaluating the performance of the anomaly detection algorithms is more complicated. On the one hand, the ground truth of anomalies is unclear because real anomalies are rare in nature. On the other hand, the anomaly detection algorithms often output an anomalous score for each object. The objects with relatively large anomalous scores are considered anomalies if they are larger than a given threshold. Setting a proper threshold for each application in advance is relatively difficult. If the threshold is set too large, true anomalies would be missed; otherwise, some objects that are not true anomalies would be mistakenly taken as potential anomalies.

In general, the following measurements have often been used to evaluate the performance of the anomaly detection methods.

(1) **Precision at *t*** ($P@t$) [48]: given a dataset D consisting of N objects, $P@t$ is defined as the proportion of the true anomalies, $A \subseteq D$, to the top t potential anomalies identified by the detection method; that is,

$$P@t = \frac{|a \in A \mid \text{rank}(a) \leq t|}{t} \tag{2}$$

It is noticeable that the value of t is difficult to set for each specific application. A commonly used strategy is to set t as the number of anomalies in the ground truth.

(2) **R-precision** [49]: this measurement is the proportion of true anomalies within the top t potential anomalies identified, where t is the number of ground truth anomalies. Since the number of true anomalies is relatively small in comparison to the size of the dataset, the value of R-precision would be very small. Thus, it contains less information.

(3) **Average precision** (AP) [50]: instead of evaluating the precision individually, this measurement refers to the mean of precision scores over the ranks of all anomaly objects:

$$AP = \frac{1}{|a|} \sum_{t=1}^{|a|} P@t. \tag{3}$$

where $P@t$ is the precision at t, that is, Eq. (2).

(4) **AUC** [4]: the receiver operating characteristic (ROC) curve is a graphical plot of the true positive rate against the false positive rate, where the true (false) positive rate represents the proportion of anomalies (inliers) ranked among the top t potential anomalies. Zimek et al. [4] noted that, for a random model, the ROC curve tends to be diagonal, while, for a good ranking model, it will output true anomalies first, leading to the area under the corresponding curve (AUC) covering all available space. Thus, the AUC is often used to numerically evaluate the performances of anomaly detection algorithms.

(5) **Correlation coefficient**: correlation coefficients, such as Spearman's rank similarity and Pearson correlation, are also taken as evaluation measurements. This kind of measurement places more emphasis on the potential anomalies ranked at the top by incorporating weights. More details about the measurements of correlation coefficients can be found in [51] and references therein.

(6) **Rank power** (RP): Both the precision and AUC criteria do not consider characteristics of anomaly ranking. Intuitively, an anomaly ranking algorithm will be regarded as more effective if it ranks true anomalies in the top and normal observations in the bottom of the list of anomaly candidates. The rank power is such a metric and evaluates the comprehensive ranking of true anomalies. The formal definition is

$$\text{RankPower} = \frac{n(n+1)}{2\sum_{i=1}^{n} R_i} \tag{4}$$

where n is the number of anomalies in the top t potential objects and R_i is the rank of the i-th true anomaly. For a fixed value of t, a larger value indicates better performance. When the t anomaly candidates are true anomalies, the rank power equals one.

4. Experimental Comparisons

As discussed above, various anomaly detection algorithms have been developed. For better understanding the characters of the detection methods, in this section we make an experimental comparison of the popular anomaly detection algorithms.

4.1. Experimental Settings. In the literature, two kinds of data, that is, synthetic and real-world datasets, were often reported to evaluate the performance of the anomaly detection methods. The former is generated under the contexts of specific

Table 2: Experimental datasets used in our experiments.

Dataset	N(A)	Attribute	Anomalies	Source
ALOI	50000(1508)	27	The 1508 of rare objects	UCI [53]
Arcene	100(44)	10000	The cancer patients	UCI [53]
Ionoshere	351(126)	32	The 'bad' class	UCI [53]
KDDCup99	48113(200)	38	The U2R class	UCI [53]
PenDigits	9868(20)	16	The fourth class	UCI [53]
Sonor	208(97)	60	The rock object	UCI [53]
WDBC	367(10)	30	The malignant class	UCI [53]
Waveform	3443(100)	21	The '0' class	UCI [53]
Ann-thyroid	7129(534)	21	The hyper and subnormal classes	ELKI [54]
Arrhythmia	450(206)	259	The arrhythmia class	ELKI [54]
HeartDisease	270(120)	13	The affected patients class	ELKI [54]
Pima	768(268)	8	The Diabetes class	ELKI [54]
SpamBase	4209(1681)	57	The non-spam class	ELKI [54]
ALLAML	38(11)	7129	The AML class	EBD [56]
DLBCL	77(19)	7129	The FL morphology class	EBD [56]
Gisette	550(50)	5000	The normal class	EBD [56]
Lung_MPM	181(31)	12533	The MPM class	EBD [56]
Ovarian	253(162)	15154	The Ovarian Cancer class	EBD [56]

constraints or conditions. Wang et al. [52] provided several synthetic datasets with anomalies for different scenarios. The real-world datasets are offered at public sources such as UCI Machine Learning Repository [53] and ELKI toolkits [54]. However, the datasets publicly available are initially used for classification purposes. Hence, they should be preprocessed, making them suitable for the anomaly detection tasks. Two strategies are frequently adopted during the preprocessing stage [55]. The classes with rare data will be regarded as anomalies and the remaining as normal ones, if they have explicitly semantically meanings. Otherwise, one of the classes will be randomly selected as the anomalies.

To make a fair comparison, our experiments were carried out on 18 real-world datasets. They were downloaded from the UCI Machine Learning Repository [53], the ELKI toolkit [54], and ELVIRA Biomedical Dataset Repository (EBD) [56]. A brief summary of the datasets is presented in Table 2, where the "N (A)" column refers to the numbers of normal objects and anomalies, respectively. We performed the preprocessed operation on the datasets as suggested in [55]. For example, the fourth class ('4') in *PenDigits* consisting of 9,868 objects was considered anomalies, while the remaining as normal objects in our experiments.

The experiments compared nine popular anomaly detection algorithms, including *k*NN (*k* Nearest Neighbours) [9], LOF (Local Anomaly Factor) [11], LoOP (Local Anomaly Probabilities) [12], ODIN (Outlier Detection using Indegree Number) [13], RBDA (Rank-Based Detection Algorithm) [6], OR (Out Rank) [17], SOD (Subspace Anomaly Degree) [7], FB (Feature Bagging) [19], and HiCS (High Contrast Subspaces) [8]. They stand for the three kinds of the anomaly detection methods as mentioned above. For example, *k*NN, ODIN, LOF, LoOP, and RBDA belong to the neighbour-based detection methods and OR and SOD are the subspace-based detection methods. FB and HiCS are the ensemble-based detection methods.

In our experiments, two metrics, that is, R-precision and AUC, were adopted to evaluate the detection algorithms. For the remaining four metrics, we have not presented here, because similar conclusions were found. The comparison experiments were conducted with the ELKI toolkit. The parameters involved within the anomaly detection algorithms were assigned to default values as recommended in the literature. The experiments were performed on a PC with 2.8 GHz of CPU clock rate and 4 GB of main memory.

4.2. Experimental Results. Table 3 provides the R-precision performance of the anomaly detection algorithms on the experimental datasets. Since the main memory was quickly consumed when RBDA, FB, and OR run on the *ALOI* and *KDDCup99* datasets, their experimental results were unavailable and presented as "/" in Table 3.

From the experimental results in Table 3, one may observe that the neighbour-based methods had relatively stable performance, while the ensemble-based methods, for example, HiCS, performed unsteadily in many cases. For instance, *k*NN and RBDA achieved relatively good performance on eight datasets. Even HiCS had worse performance on four of them, for example, *PenDigits*, *KDDCup99*, *Ann-thyroid,* and *DLBCL*, but it achieved the highest R-precisions on *Waveform*, *WDBC,* and *Ovarian*. The reason is that the ensemble-based detection methods tend to be sensitive to the size of datasets subsampled from the original ones. Since *OR* is heavily dependent on the quantities of feature subspaces, it obtained the highest values on *Ann-thyroid* and the lowest values on *Sonar*, *Waveform*, *Arrhythmia,* and *Spambase*. For the high-dimensional datasets, that is, *Arcene, ALLAML, DLBCL, Gisette, Lung_MPM,* and *Ovarian*, *k*NN,

Table 3: R-precisions of the anomaly detection algorithms where k=7 for the neighbours.

Dataset	*k*NN	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	0.16	0.24	0.20	0.22	/	0.06	0.21	/	/
Ionoshere	0.65	0.57	0.46	0.65	0.88	0.15	0.69	0.77	0.69
KDDCup99	0.09	0.24	0.15	0.20	/	/	0.44	/	/
PenDigits	0.01	0.05	0.05	0.05	0.01	0.01	0.05	0.01	0.01
Sonar	0.46	0.55	0.55	0.63	0.53	0.45	0.52	0.49	0.57
WDBC	0.30	0.30	0.40	0.30	0.41	0.6	0.60	0.7	0.70
Waveform	0.07	0.05	0.06	0.04	0.10	0.03	0.06	0.13	0.21
Arrhythmia	0.67	0.64	0.66	0.66	0.73	0.60	0.65	0.65	0.61
Ann-thyroid	0.05	0.04	0.04	0.04	0.06	0.14	0.01	0.04	0.01
HeartDisease	0.57	0.53	0.49	0.56	0.48	0.49	0.52	0.43	0.50
Pima	0.50	0.45	0.39	0.52	0.41	0.39	0.52	0.35	0.46
SpamBase	0.52	0.43	0.40	0.45	0.38	0.38	0.43	0.39	0.48
Arcene	0.52	0.29	0.39	0.42	0.43	0.45	0.48	0.40	0.45
ALLAML	0.40	0.36	0.36	0.36	0.39	0.36	0.36	0.36	/
DLBCL	0.21	0.09	0.21	0.21	0.24	0.21	0.21	0.21	0.16
Gisette	0.14	0.12	0.10	0.14	0.15	0.14	0.16	0.16	0.10
Lung_MPM	0.52	0.29	0.39	0.42	0.54	0.45	0.48	0.49	0.45
Ovarian	0.54	0.59	0.56	0.60	0.61	0.60	0.57	0.53	0.62

RBDA, SOD, and OR had good performance, where OR and SOD were more stable. Indeed, these contain many irrelevant features, which makes those subspace-based methods more effective. It can also be observed that RBDA was better than *k*NN and ODIN, for RBDA took the neighbour ranking, instead of the distances, into account which is more suitable for the high-dimensional datasets.

The compared algorithms, except OR, take *k*NN as their baseline. As we know, *k*NN heavily relies on the number of neighbours *k*. To reveal the impact of *k* on the performance, we performed a comparison experiment among these methods with different *k* values. Tables 4 and 5 show the AUC scores of the anomaly detection algorithms with *k*=10 and *k*=50, respectively. Since the experimental results on the high-dimensional datasets (i.e., *Arcene, ALLAML, DLBCL, Gisette, Lung_MPM,* and *Ovarian*) were still unavailable after three hours' running, they were not provided in Table 5.

According to the results, we know that the detection performance of the comparison algorithms was heavily dependent on the number of neighbours and varied greatly when *k* assigned different values. To further illustrate this fact, we conducted additional experiments by performing the detection algorithms on *Arrhythmia*, *Waveform*, and *WDBC* with *k* varying from 10 to 50. The experimental results are illustrated as Figure 1.

As shown in Figure 1, the performance of RBDA, FB, and SOD was relatively stable, although they took use of *k*NN as their baselines. In fact, SOD exploits *k*NN to obtain the relative subspace information, while FB ensembles all informative features found by *k*NN. As a result, *k* had less impact on them. On the other hand, *k*NN, ODIN, LoOP, and LOF heavily relied on the values of *k*. For example, the AUC values from ODIN varied greatly on all three datasets with the different values of *k*. HiCS had unsteady performance in many cases. For instance, it was less affected by *k* on WDBC, while sensitive to *k* on *Arrhythmia*. The reason is that, in our experiments, the basis detector of HiCS was also *k*NN, leading to its performance relying on *k*, although it is an ensemble anomaly detection algorithm.

Another interesting fact is that, on the *WDBC* and *Waveform* datasets, the AUC values of the compared algorithms varied greatly. Indeed, the ratios of anomalies to normal objects within these two datasets are relatively small (2.7% and 2.9% for *WDBC* and *Waveform*, respectively). Consequently, the anomaly detection algorithms were more sensitive to *k*. In contrast, on the datasets with high anomaly proportions, for example, *Arrhythmia* (45.7% anomalies), the AUC scores of the anomaly detection algorithms were less sensitive to *k*. Similar situations can be found for the other datasets. Due to the limitation of space, they will not be presented here one by one.

Computational efficiency is another important aspect for the practical applications of the anomaly detection methods. We carried out an additional experiment to compare the computational efficiencies of the anomaly detection algorithms. Table 6 records the elapsed time (s) of the anomaly detection algorithms on the experimental datasets.

The elapsed time in Table 6 shows that the neighbour-based detection methods, using the metrics of both distances (e.g., *k*NN and ODIN) and densities (e.g., LOF and LoOP), had relatively higher efficiencies. However, the ensemble-based detection methods, especially HiCS, took too much time to detect anomalies. As a matter of fact, they construct lots of individual detectors before identifying outliers. For the subspace-based detection algorithms, their efficiencies are dependent on the techniques adopted. For example, SOD, which exploits neighbours to explore relative subspaces, is more efficient than OR.

Table 4: AUC of the anomaly detection algorithms with k=10 for the neighbours.

Dataset	kNN	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	0.66	0.80	0.78	0.80	/	0.57	0.72	/	/
Ionoshere	0.49	0.51	0.57	0.71	0.89	0.24	0.76	0.88	0.80
KDDCup99	0.70	0.60	0.59	0.81	/	/	0.91	/	/
PenDigits	0.90	0.88	0.90	0.88	0.56	0.47	0.91	0.80	0.81
Sonar	0.60	0.60	0.61	0.66	0.60	0.49	0.51	0.57	0.59
WDBC	0.64	0.80	0.69	0.76	0.89	0.96	0.90	0.94	0.98
Waveform	0.53	0.52	0.48	0.54	0.70	0.57	0.63	0.73	0.73
Arrhythmia	0.75	0.68	0.73	0.72	0.73	0.68	0.71	0.73	0.69
Ann-thyroid	0.52	0.50	0.50	0.52	0.69	0.54	0.47	0.72	0.54
HeartDisease	0.52	0.48	0.46	0.59	0.52	0.55	0.61	0.52	0.46
Pima	0.59	0.49	0.49	0.62	0.58	0.54	0.65	0.50	0.54
SpamBase	0.58	0.50	0.51	0.53	0.47	0.46	0.55	0.48	0.52
Arcene	0.46	0.46	0.45	0.46	0.46	0.52	0.47	0.40	0.49
ALLAML	0.71	0.66	0.69	0.69	0.70	0.70	0.72	0.69	/
DLBCL	0.40	0.39	0.40	0.42	0.40	0.41	0.36	0.41	0.40
Gisette	0.56	0.55	0.58	0.56	0.57	0.58	0.71	0.58	0.44
Lung_MPM	0.80	0.63	0.73	0.73	0.71	0.69	0.71	0.73	0.75
Ovarian	0.32	0.38	0.38	0.46	0.43	0.43	0.37	0.38	0.44

Table 5: AUC of the anomaly detection algorithms with k=50 for the neighbours, where the performance on *Arcene*, *ALLAML*, *DLBCL*, *Gisette*, *Lung_MPM,* and *Ovarian* was not given, for it was still unavailable after three hours' running.

Dataset	kNN	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	0.59	0.75	0.74	0.77	/	0.57	0.71	/	/
Ionoshere	0.48	0.50	0.55	0.63	0.89	0.24	0.77	0.86	0.75
KDDCup99	0.67	0.66	0.62	0.65	/	/	0.89	/	/
PenDigits	0.65	0.66	0.76	0.84	0.92	0.47	0.88	0.96	0.86
Sonar	0.56	0.55	0.55	0.58	0.59	0.49	0.56	0.54	0.61
WDBC	0.61	0.51	0.61	0.50	0.90	0.96	0.90	0.92	0.98
Waveform	0.48	0.48	0.48	0.51	0.73	0.57	0.63	0.73	0.74
Arrhythmia	0.75	0.71	0.74	0.73	0.74	0.68	0.72	0.75	0.61
Ann-thyroid	0.51	0.52	0.53	0.51	0.66	0.54	0.47	0.65	0.52
HeartDisease	0.53	0.47	0.46	0.57	0.56	0.55	0.60	0.64	0.46
Pima	0.57	0.53	0.52	0.62	0.62	0.54	0.65	0.62	0.61
SpamBase	0.63	0.50	0.52	0.53	0.50	0.46	0.55	0.39	0.54

5. Conclusion

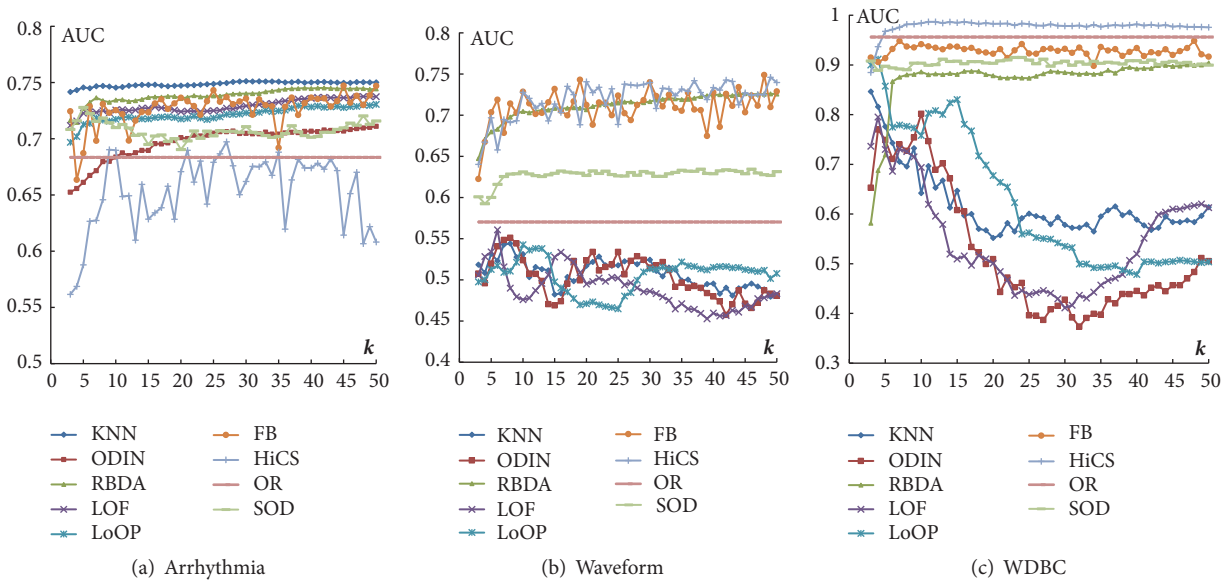
The data collected from real-world applications are becoming larger and larger in size and dimension. As the dimensionality increases, the data objects become sparse, resulting in identifying anomalies being more challenging. Besides, the conventional anomaly detection methods cannot work effectively and efficiently. In this paper, we have discussed typical problems of anomaly detection associated with the high-dimensional and mixed-type data and briefly reviewed the techniques of anomaly detection. To offer a better understanding of the anomaly detection techniques for practitioners, we conducted extensive experiments on publicly available datasets to evaluate the typical and popular anomaly detection methods. Although the progresses of anomaly detection for the high-dimensional and mixed-type data have been achieved to some extent, there are also several open issues shown as follows that further need to be addressed:

(1) The traditional distance metrics in the neighbour-based methods cannot work very well for the high-dimensional data because of the equidistant characteristics. The mixed-type features make anomaly detection more difficult. Introducing effective distance metrics for the high-dimensional and mixed-type data is necessary.

(2) The neighbour-based anomaly detection algorithms are sensitive to nearest neighbours selected for the models. Determining the right number of neighbours is a challenging issue for the neighbour-based methods.

Table 6: Time cost (s) of the anomaly detection algorithms, where k=10.

Dataset	*k*NN	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	163.3	131.2	129.8	134.9	/	1650.8	273.3	/	/
Ionoshere	0.02	0.03	0.03	0.05	0.04	2.10	0.05	1.07	171.70
KDDCup99	164.1	166.9	168.4	168	/	/	325.1	/	/
PenDigits	2.13	2.07	2.11	2.20	2.20	414.52	13.8	156.20	2549.47
Sonar	0.02	0.02	0.02	0.02	0.04	1.318	0.04	0.125	17.37
WDBC	0.09	0.05	0.09	0.05	0.11	2.51	0.08	0.34	58.05
Waveform	0.13	0.83	0.81	0.70	0.16	400.09	3.78	60.28	5924.89
Arrhythmia	0.27	0.16	0.17	0.17	0.30	35.68	0.11	2.45	64.74
Annthyroid	1.42	1.46	1.46	1.47	1.50	349.59	4.93	261.23	4514.92
HeartDisease	0.02	0.01	0.03	0.02	0.04	0.45	0.03	0.34	126.21
Pima	0.04	0.04	0.04	0.04	0.07	2.55	0.37	0.51	95.78
SpamBase	2.32	3.85	3.73	2.22	2.80	589.59	4.56	151.48	10453.15
Arcene	0.59	0.60	0.60	0.60	0.62	1158.85	0.70	19.44	7581.04
ALLAML	0.06	0.07	0.06	0.07	0.08	3.73	0.10	0.42	/
DLBCL	0.26	0.26	0.27	0.261	0.32	28.55	0.36	1.69	3496.79
Gisette	9.81	9.06	9.08	9.12	10.80	56754	9.65	67.87	11370.13
Lung_MPM	2.41	2.46	2.46	2.47	2.88	850.21	2.81	17.32	96292.66
Ovarian	5.83	5.79	5.80	5.80	6.76	1521.78	6.26	40.76	1367821.09



(a) Arrhythmia (b) Waveform (c) WDBC

Figure 1: AUC of the anomaly detection algorithms with different *k* varying from 3 to 50 on *Arrhythmia*, *Waveform,* and *WDBC*.

(3) The subspace-based and ensemble-based methods have relatively good performance if the diversity of the subspaces or base learners is large. For these kinds of anomaly detection methods, how to choose the right subspaces or base learners, as well as their quantities and their combining strategies, is still and open issue.

(4) Since anomalies are relatively rare and the ground truth is often unavailable in real scenarios, how to effectively and comprehensively evaluate the detection performance is also a challenging issue.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Funding

This work was supported by the National Natural Science Foundation (NSF) of China (61871350,61572443); the Natural Science Foundation of Zhejiang Province of China (LY14F020019); and Shanghai Key Laboratory of Intelligent Information Processing, Fudan University (IIPL-2016-001).

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

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

[3] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2014.

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

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

[6] H. Huang, K. Mehrotra, and C. K. Mohan, "Rank-based outlier detection," *Journal of Statistical Computation and Simulation*, vol. 83, no. 3, pp. 518–531, 2013.

[7] H. P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data," in *Proceedings of the Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 831–838, Springer-Verlag, 2009.

[8] F. Keller, E. Müller, and K. Böhm, "HiCS: High contrast subspaces for density-based outlier ranking," in *Proceedings of the IEEE 28th International Conference on Data Engineering, ICDE 2012*, pp. 1037–1048, USA, April 2012.

[9] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000.

[10] F. Angiulli and C. Pizzuti, "Fast Outlier Detection in High Dimensional Spaces," in *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 15–26, Springer-Verlag, Heidelberg, Berlin, Germany, 2002.

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

[12] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities," in *Proceedings of the ACM 18th International Conference on Information and Knowledge Management (CIKM '09)*, pp. 1649–1652, ACM Press, November 2009.

[13] H. Ville, I. Karkkainen, and P. Franti, "Outlier Detection Using k-Nearest Neighbour Graph," in *Proccedings of the IEEE International Conference on Pattern Recognition*, vol. 3, pp. 330–433, 2004.

[14] J. Zhang, Y. Jiang, K. H. Chang, S. Zhang, J. Cai, and L. Hu, "A concept lattice based outlier mining method in low-dimensional subspaces," *Pattern Recognition Letters*, vol. 30, no. 15, pp. 1434–1439, 2009.

[15] J. Zhang, X. Yu, Y. Li, S. Zhang, Y. Xun, and X. Qin, "A relevant subspace based contextual outlier mining algorithm," *Knowledge-Based Systems*, vol. 99, no. 72, pp. 1–9, 2016.

[16] J. K. Dutta, B. Banerjee, and C. K. Reddy, "RODS: Rarity based Outlier Detection in a Sparse Coding Framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 483–495, 2016.

[17] E. Müller, I. Assent, U. Steinhausen, and T. Seidl, "OutRank: Ranking outliers in high dimensional data," in *Proceedings of the 2008 - IEEE 24th International Conference on Data Engineering Workshop, ICDE'08*, pp. 600–603, Mexico, April 2008.

[18] E. Müller, M. Schiffer, and T. Seidl, "Adaptive outlierness for subspace outlier ranking," in *Proceedings of the 19th International Conference on Information and Knowledge Management and Co-located Workshops, CIKM'10*, pp. 1629–1632, Canada, October 2010.

[19] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceedings of the KDD-2005: 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 157–166, USA, August 2005.

[20] B. Van Stein, M. Van Leeuwen, and T. Back, "Local subspace-based outlier detection using global neighbourhoods," in *Proceedings of the 4th IEEE International Conference on Big Data, Big Data 2016*, pp. 1136–1142, USA, December 2016.

[21] A. Zimek, M. Gaudet, R. J. G. B. Campello, and J. Sander, "Subsampling for efficient and effective unsupervised outlier detection ensembles," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013*, pp. 428–436, USA, August 2013.

[22] J. R. Pasillas-Diaz and S. Ratte, "Bagged subspaces for unsupervised outlier detection," *International Journal of Computational Intelligence*, vol. 33, no. 3, pp. 507–523, 2017.

[23] A. Ghoting, M. E. Otey, and S. Parthasarathy, "LOADED: Link-based outlier and anomaly detection in evolving data sets," in *Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM 2004*, pp. 387–390, UK, November 2004.

[24] A. Koufakou and M. Georgiopoulos, "A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes," *Data Mining and Knowledge Discovery*, vol. 20, no. 2, pp. 259–289, 2010.

[25] K. Zhang and H. Jin, "An effective pattern based outlier detection approach for mixed attribute data," in *AI 2010: Advances in Artificial Intelligence*, vol. 6464 of *Lecture Notes in Computer Science*, pp. 122–131, Springer, Berlin, Germany, 2010.

[26] Y.-C. Lu, F. Chen, Y. Wang, and C.-T. Lu, "Discovering anomalies on mixed-type data using a generalized Student-t based approach," *Expert Systems with Applications*, vol. 28, no. 10, pp. 1–10, 2016.

[27] K. Do, T. Tran, D. Phung, and S. Venkatesh, "Outlier detection on mixed-type data: an energy-based approach," in *Advanced Data Mining and Applications*, pp. 111–125, Springer International Publishing, Cham, switzerland, 2016.

[28] H. Huang, K. Mehrotra, and C. K. Mohan, "Outlier detection using modified-ranks and other variants," Electrical Engineering and Computer Science 72, 2011, https://surface.syr.edu/eecs_techreports/72/.

[29] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1369–1382, 2015.

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

[31] L. Zhang, Z. He, and D. Lei, "Shared nearest neighbors based outlier detection for biological sequences," *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 12, pp. 1–10, 2012.

[32] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.

[33] J. Ha, S. Seok, and J.-S. Lee, "A precise ranking method for outlier detection," *Information Sciences*, vol. 324, pp. 88–107, 2015.

[34] C. C. Aggarwal, "High dimensional outlier detection: the subspace method," in *Outlier Analysis*, pp. 135–167, Springer, New York, NY, USA, 2013.

[35] J. Zhang, S. Zhang, K. H. Chang, and X. Qin, "An outlier mining algorithm based on constrained concept lattice," *International Journal of Systems Science*, vol. 45, no. 5, pp. 1170–1179, 2014.

[36] C. C. Aggarwal and S. Yu, *An Effective and Efficient Algorithm for High-Dimensional Outlier Detection*, Springer-Verlag, New York, NY, USA, 2005.

[37] E. Muller, M. Schiffer, and T. Seidl, "Statistical selection of relevant subspace projections for outlier ranking," in *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE 2011*, pp. 434–445, Germany, April 2011.

[38] M. A. Stephens, "Use of the kolmogorov-smirnov, cramér-von mises and related statistics without extensive tables," *Journal of the Royal Statistical Society: Series B*, vol. 32, no. 1, pp. 115–122, 1970.

[39] A. Zimek, R. J. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: challenges and research questions," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 1, pp. 11–22, 2014.

[40] C. C. Aggarwal and S. Sathe, "Theoretical Foundations and Algorithms for Outlier Ensembles," *ACM SIGKDD Explorations Newsletter*, vol. 17, no. 1, pp. 24–47, 2015.

[41] 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*, vol. 5981, pp. 368–383, Springer, Berlin, Germany, 2010.

[42] A. Giacometti and A. Soulet, "Frequent pattern outlier detection without exhaustive mining," *Advances in Knowledge Discovery and Data Mining*, pp. 196–207, 2016.

[43] Z. He, X. Xu, Z. Huang, and S. Deng, "FP-outlier: Frequent pattern based outlier detection," *Computer Science and Information Systems*, vol. 2, no. 1, pp. 103–118, 2005.

[44] M. E. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets," *Data Mining and Knowledge Discovery*, vol. 12, no. 2-3, pp. 203–228, 2006.

[45] S. Wu and S. Wang, "Information-theoretic outlier detection for large-scale categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 3, no. 25, pp. 589–602, 2013.

[46] A. Koufakou, J. Secretan, and M. Georgiopoulos, "Non-derivable itemsets for fast outlier detection in large high-dimensional categorical data," *Knowledge and Information Systems*, vol. 29, no. 3, pp. 697–725, 2011.

[47] M. Bouguessa, "A practical outlier detection approach for mixed-attribute data," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8637–8649, 2015.

[48] N. Craswell, "Precision at n," in *Encyclopaedia of Database Systems*, L. Liu and M. Ozsu, Eds., pp. 2127-2128, Springer, Berlin, Germany, 2009.

[49] N. Craswell, "R-precision," in *Encyclopaedia of Database Systems*, L. Liu and M. Ozsu, Eds., p. 2453, Springer, Berlin, Germany, 2009.

[50] E. Zhang and Y. Zhang, "Average precision," in *Encyclopaedia of Database Systems*, L. Liu and M. Ozsu, Eds., pp. 192-193, Springer, Berlin, Germany, 2009.

[51] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*, pp. 1047–1058, USA, April 2012.

[52] X. Wang, X. L. Wang, Y. Ma, and D. M. Wilkes, "A fast MST-inspired kNN-based outlier detection method," *Information Systems*, vol. 48, pp. 89–112, 2015.

[53] "UCI Machine Learning Repository," 2007, http://archive.ics.uci.edu/ml/.

[54] "ELKI," 2016, https://elki-project.github.io/releases/.

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

[56] "ELVIRA Biomedical DataSet Repository," 2005, http://leo.ugr.es/elvira/DBCRepository/.