A Taxonomy Framework for Unsupervised Outlier Detection Techniques for Multi-Type Data Sets

Yang Zhang, Nirvana Meratnia, Paul Havinga
Department of Computer Science,
University of Twente, P.O.Box 217 7500AE,
Enschede, The Netherlands

The term "outlier" can generally be defined as an observation that is significantly different from the other values in a data set. The outliers may be instances of error or indicate events. The task of outlier detection aims at identifying such outliers in order to improve the analysis of data and further discover interesting and useful knowledge about unusual events within numerous applications domains. In this paper, we report on contemporary unsupervised outlier detection techniques for multiple types of data sets and provide a comprehensive taxonomy framework and two decision trees to select the most suitable technique based on data set. Furthermore, we highlight the advantages, disadvantages and performance issues of each class of outlier detection techniques under this taxonomy framework.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*data mining*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Outlier, outlier detection, data mining

1. INTRODUCTION

Data mining, as a powerful knowledge discovery tool, aims at modelling relationships and discovering hidden patterns in large databases [1]. Among four typical data mining tasks, *outlier detection* is the closest to the initial motivation behind data mining than predictive modelling, cluster analysis and association analysis [2]. Outlier detection has been a widely researched problem in several knowledge disciplines, including *statistics*, *data mining* and *machine learning*. It is also known as anomaly detection, deviation detection, novelty detection and exception mining in some literature [3]. Being called differently, all these definitions aim at identifying instances of unusual behavior when compared to the majority of observations.

Coming across various definitions of an outlier, it seems that no universally accepted definition exists. Two classical definitions of an outlier include Hawkins [4] and Barnett and Lewis [5]. According to the former, *"an outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism"*, where as the latter defines an outlier is *"an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data"*. The term "outlier" can generally be defined as an observation that is significantly different from the other values in a data set. However, as it will be presented in section 3, the notion of outliers may even differ from one outlier detection technique to another.

Outliers often occur due to the following reasons, which make occurrence of an outlier typically being an indication of an *error* or an *event* [6].

— *Error*. This sort of outliers are also known as anomalies, discordant observations, exceptions, faults, defects, aberrations, noise, damage or contaminants. They may occur because of human errors, instrument errors, mechanical faults or change in the environment. Due to the fact that such outliers reduce the quality of data analysis and so may lead to erroneous results, they need to be identified and immediately discarded.

— *Event*. As stated in [4], outliers may be generated by a *"different mechanism"*, which indicates that this sort of outliers belong to unexpect patterns that do not conform to normal behavior and may include interesting and useful information about rarely occurring events within numerous application domains. Therefore, it is worthwhile that such outliers would be identified for further investigation.

Over the years, outlier detection has been widely applied for numerous applications domains such as those described below:

— *Fraud detection* [7]. The purchasing behavior of people who steal credit cards may be different from that of the owners of the cards. The identification of such buying pattern changes could effectively prevent thieves from a long period of fraud activity. Similar approaches can also be used for other kinds of commercial fraud such as in mobile phones, insurance claim, financial transactions etc [7].

— *Intrusion detection* [8]. Frequent attacks on computer systems may result in systems being disabled, even completely collapsing. The identification of such intrusions could find out malicious programs in computer operating system and also detect unauthorized access with malicious intentions to computer network systems and so effectively keep out hackers.

— *Environmental monitoring* [9]. Many unusual events that occur in the natural environment such as a typhoon, flooding, drought and fire, often have an adverse impact on the normal life of human beings. The identification of certain atypical behaviors could accurately predict the likelihood of these phenomena and allow people to take effective measures on time.

— *Medical and public health* [10]. Patient records with unusual symptoms or test results may indicate potential health problems for a particular patient. The identification of such unusual records could distinguish instrumentation or recording errors from whether the patient really has potential diseases and so take effective medical measures in time.

— *Localization and tracking* [11]. Localization refers to the determination of the location of an object or a set of objects. The collection of raw data can be used to calibrate and localize the nodes of a network while simultaneously tracking a moving target. It is a known fact that raw data may contain error, which makes localization results not accurate and useful. Filtering such erroneous data could improve the estimation of the location of objects and make tracking easier.

— *Logistics and transportation* [12]. Logistics refers to manage and control the flow of products from the source of production to the destination. It is very essential to ensure product safety and product reliability issues during this process. Tracking and tracing shipments could find out potential exceptions, e.g., inappropriate quantity and quality of the product, and notify all trading partners in time.

Based on these real-life applications, it can clearly be seen that outlier detection is a quite critical part of any data analysis. In the detection of outliers, there is a universally accepted assumption that the number of anomalous data is considerably smaller than normal data in a data set. Thus, a straightforward approach to identify outliers is to construct a profile of the normal behaviors of the data and then use certain measure methods to calculate the degree to which data deviate from the profile in a data set. Those instances that significantly deviate from the profile are declared as outliers [1]. However, existing methods using *pre-labelled data* to build a normal model in a training phase before detecting outliers are very challenging since not all possible normal behaviors have been encompassed within the normal model. For example, a data stream refers to a large number of data which continuously evolves with the time. This may cause that the normal model built in a particular time instant is invalid in consequent time instants. In this paper, we describes *unsupervised* outlier detection techniques that require no labelled training data.

Markou and Singh [13] and [14] present an extensive review of novelty detection techniques based on statistical and neural network approaches, respectively. However, they do not classify outlier detection techniques based on different types of data sets. Hodge and Austin [3] address outlier detection methodologies from three fields of computing, i.e., statistics, neural networks and machine learning. Outlier detection techniques presented in these surveys focus only on simple data sets in which the data usually is represented by low-dimensional real-valued attributes. Quite often, these techniques are not suitable for complex data sets such as high dimensional, mixed-type attributes, sequence, spatial, streaming and spatio-temporal nature of data sets. To the best of our knowledge, the most extensive survey on more complex outlier detection techniques is the work of Chandola et al. [100], in which authors classify outlier detection techniques in terms of various application domains and several knowledge disciplines.

In this paper, we focus on performance evaluation of different outlier detection techniques with respect to type of data sets they handle. Our work goes beyond existing surveys because we provide a comprehensive taxonomy framework and two decision trees to choose suitable techniques for specific application domains and data set, and also introduce a through performance evaluation of each class of outlier detection techniques.

The contributions of this paper are the following, we:

— present a comprehensive taxonomy framework for contemporary outlier detection techniques based on multiple types of data sets.

— discuss the key characteristic of the current unsupervised outlier detection techniques for multiple types of data sets.

— provide a through performance evaluation of each class of outlier detection techniques.

— introduce two decision trees to choose suitable outlier detection techniques based on domains of applications and types of data sets, respectively.

The rest of this paper is organized as follows. In Section 2, we discuss classification criterion of general-purpose outlier detection techniques. In Section 3, we present a comprehensive taxonomy framework for contemporary outlier detection

techniques. The most commonly used outlier detection techniques for the simple data set are presented in Section 4. The novel outlier detection techniques for complex data sets with specific data semantics are presented in Section 5. In Section 6, we present two decision trees based on applications and types of data sets. We conclude the paper in Section 7.

2. CLASSIFICATION CRITERION

As mentioned earlier, various outlier detection approaches work differently for particular data sets, in terms of the accuracy and execute time. No single universally applicable or generic outlier detection approach exists [3]. Thus, it is very critical to design an appropriate outlier detection approach for a given data set. In this section, we summarize several important aspects related to *general-purpose* outlier detection techniques and commonly used evaluation metrics. Furthermore, these aspects will be used as metrics to compare characteristics of different outlier detection techniques in section 6.

2.1 Characteristics of Outliers

2.1.1 *Type of Detected Outliers: Global vs Local.* Outliers can be identified as either global or local outliers. A global outlier is an anomalous data point with respect to all other points in the whole data set, but may not with respect to points in its local neighborhood. A local outlier is a data point that is significantly different with respect to other points in its local neighborhood, but may not be an outlier in a global view of the data set.

2.1.2 *Degree of Being an Outlier: Scalar vs Outlierness.* A data point can be considered as an outlier in two manners, scalar (binary) or outlierness. The scalar fashion is that the point is either an outlier or not. On the other hand, the outlierness fashion provides the degree of which the point is an outlier when compared to other points in a data set. This outlierness is also known as *anomaly score* or *outlier score* [15], which usually can be calculated by using specific measure methods.

2.1.3 *Dimension of Detected Outliers: Univariate vs Multivariate.* Whether a data point is an outlier is determined by the values of its attributes. A univariate data that has a single attribute can be detected as an outlier only based on the fact that a single attribute is anomalous with respect to that of other data. On the other hand, a multivariate data that has multiple attributes may be identified as an outlier since some of its attributes together have anomalous values, even if none of its attributes individually has an anomalous value. Thus, designing those techniques for detecting multivariate outliers become more complicated.

2.1.4 *Number of Detected Outliers at Once: One vs Multiple.* Outlier detection techniques can be designed to identify different number of outliers at a time. In some techniques, one outlier is identified and removed at a time, then the procedure will be repeated until no outliers are detected. These techniques may be subject to the problem of missing some real outliers during the iteration. On the other hand, for other techniques, they can identify a collection of outliers at once. However, these techniques may cause some normal data to be declared as outliers in operation.

2.2 Characteristics of Outlier Detection Approaches

2.2.1 *Use of Pre-labelled Data: Supervised vs Unsupervised.* Outlier detection approaches can generally be classified into three basic categories, i.e., supervised, unsupervised and semi-supervised learning approaches. This categorization is based on the degree of using pre-defined labels to classify normal or abnormal data [15].

— *Supervised learning approach.* These approaches initially require the learning of a normality and an abnormality models by using pre-labelled data, and then classify a new data point as normal or abnormal depending on which model the data point fits into. These supervised learning approaches usually are applied for many fraud detection and intrusion detection applications. However, they have two major drawbacks, i.e., pre-labelled data is not easy to obtain in many real-life applications, and also new types of rare events may not be included in pre-labelled data.

— *Unsupervised learning approach.* These approaches can identify outliers without the need of pre-labelled data. For example, distributed-based methods identify outliers based on a standard statistical distribution model. Similarly, distance-based methods identify outliers based on the measure of full dimensional distance between a point and its nearest neighbors. Compared to supervised learning approaches, these unsupervised learning approaches are more *general* because they do not need pre-labelled data that are not available in many practical applications. In this paper, we will focus on unsupervised learning approaches.

— *Semi-supervised learning approach.* Unlike supervised learning approaches, these semi-supervised learning approaches only require training on pre-labelled normal data to learn a boundary of normality, and then classify a new data point as normal or abnormal depending on how well the data point fits into the normality model. These approaches require no pre-labelled abnormal data, but suffer from the same problem as supervised learning approaches, i.e., a set of representative normal data difficult to obtain in many real-life applications.

2.2.2 *Use of Parameters of Data Distribution: Parametric vs Non-parametric.* Unsupervised learning approaches can be further grouped into three categories, i.e., parametric, non-parametric and semi-parametric methods, on the basis of the degree of using the parameters of the underlying data distribution [13].

— *Parametric method.* These methods assume that the whole data can be modelled to one standard statistical distribution (e.g., the normal distribution), and then directly calculate the parameters of this distribution based on means and covariance of the original data. A point that deviates significantly from the data model is declared as an outlier. These methods are suitable for situations in which the data distribution model is a priori known and parameter settings have been previously determined. However, in many practical situations, a priori knowledge of the underlying data distribution is not always available and also it may not be a simple task to compute the parameters of the data distribution.

— *Non-parametric method.* These methods make no assumption on the statistic properties of data and instead identify outliers based on the full dimensional distance measure between points. Outliers are considered as those points that

are distant from their own neighbors in the data set. These methods also use some user-defined parameters ranging from the size of local neighborhood to the threshold of distance measure. Compared to parametric methods, these non-parametric methods are more *flexible* and *autonomous* due to the fact that they require no data distribution knowledge. However, they may have expensive time complexity, especially for high dimensional data sets. Also, the choice of appropriate values for user-defined parameters is not really easy.

— *Semi-parametric method.* These methods do not assume a standard data distribution for data, but instead map the data into a trained network model or a feature space to further identify if these points deviate from the trained network model or are distant from other points in the feature space, on the basis of some *classification* techniques such as neural network and support vector machine. In this paper, some novel *unsupervised* neural network and support vector machine approaches for outlier detection will be further described.

2.3 Type of Data Set

As addressed before, various outlier detection approaches work differently for different sets of data types. Here, we describe several common types of data sets based on the characteristics and attributes of data. They are divided into *simple* and *complex* data sets, of which the latter can be further categorized into high dimensional, mixed-type attributes, sequence, spatial, streaming and spatio-temporal data sets based on different semantics of data. These complex data sets pose the significant challenges to the outlier detection problem.

2.3.1 *Simple Data Set.* The simple data set belongs to a commonly used data set, where the data has no complex semantics and usually is represented by low-dimensional real-valued ordering attributes. Most existing outlier detection techniques are applicable for such simple data sets.

2.3.2 *High Dimensional Data Set.* This data set contains a large number of data and each data point also has a large number of attributes. As stated before, detecting multivariate outliers is more complicated, thus many outlier detection techniques may be susceptible to the problem of *the curse of dimensionality* [16] in high-dimensional data sets, especially high computation complexity and no sufficient similarity measures.

2.3.3 *Mixed-Type Attributes Data Set.* In some practical applications, the data contains the mixture of continuous (numeric) and categorical attributes. The latter usually has non-numeric and partial ordering values, e.g., city names, or type of diseases. This makes it very difficult to measure the similarity between points by commonly used measure methods. Also, the performance of detecting outliers may be influenced if the categorical data is simply disregarded.

2.3.4 *Sequence Data Set.* In the sequence data set, the data is naturally represented as a sequence of individual entities, such as symbols or letters. Also, the data has not the same length and no priori known distribution. For example, a composition of DNA is a sequence from an alphabet set {A, G, C, T}. This makes it very difficult to define a standard notion of similarity to measure structural differences

between two sequences.

2.3.5 *Spatial Data Set.* Attributes of spatial data set are distinguished as spatial and non-spatial attributes. Spatial attributes contain location, shape, directions and other geometric or topological information. They can determine spatial neighborhoods in terms of spatial relationships such as distance or adjacency. On the other hand, non-spatial attributes include the intrinsical information of data characteristic, which are used to compare and distinguish spatial points in the spatial neighborhood. This requires that outlier detection techniques can consider the property of spatial correlation of data during the detection of outliers.

2.3.6 *Streaming Data Set.* A data stream is a large data that is arriving continuously and fast in the ordered sequence. They usually are unlimited in size and occur in many real-time applications. For example, a huge amount of data of the average daily temperature are collected to the base station in wireless sensor networks continually. Thus, an efficient outlier detection technique is required to deal with the data streams in an online fashion.

2.3.7 *Spatio-Temporal Data Set.* Due to the fact that many geographic phenomena are evolving over time, the temporal aspect and spatial-temporal relationships existing among spatial data also should be considered in detecting outliers for real-life applications, e.g., geographic information systems (GIS), robotics, mobile computing, traffic analysis etc.

2.4 Evaluation Methods

2.4.1 *Detection Rate, False Alarm Rate and ROC Curves.* The effectiveness of outlier detection techniques can typically be evaluated depending on how many outliers are correctly identified and also how many normal data are incorrectly considered as outliers, the latter of which is also known as *false alarm rate*. The receiver operating characteristic (ROC) curves [17] shown in a 2-D graph usually is used to represent the trade-off between the detection rate and false alarm rate. Figure 1 illustrates an example of ROC curve. The effectiveness of outlier detection techniques is desired to maintain a high detection rate while keeping the false alarm rate low [1].

2.4.2 *Computational Complexity.* The efficiency of outlier detection techniques can be evaluated by the computational cost, which is known as time & space complexity. Also, the efficient outlier detection techniques should be scalable to a large and high dimensional data set. In addition, the amount of memory occupation required to execute outlier detection techniques can be viewed as an important performance evaluation metrics.

2.4.3 *User-Defined Parameter.* User-defined parameters are quite critical to the performance of outlier detection techniques in terms of effectiveness and efficiency. These parameters are usually used to define the size of local neighborhood of a point or the threshold of similarity measure. However, the choice of appropriate parameters is not really easy. Thus, the minimal use of user-defined parameters can enhance the applicability of outlier detection techniques.

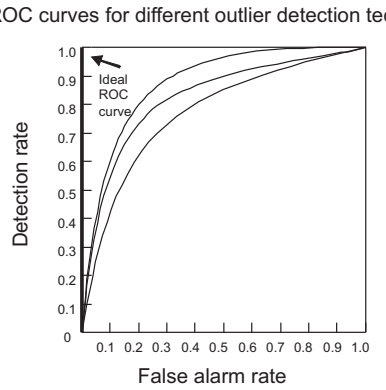

Fig. 1. ROC Curves for different detection techniques

3. TAXONOMY FRAMEWORK FOR OUTLIER DETECTION TECHNIQUES

In this section, we present a comprehensive taxonomy framework for current outlier detection techniques. Also, we briefly describe each class of outlier detection techniques under this taxonomy framework. In addition, a collection of prevalent definitions of outliers are presented with respect to different outlier detection techniques.

3.1 Taxonomy Framework

In this paper, we classify non-supervised outlier detection techniques based on the semantics of input data, as shown in Figure 2. Input data can be classified as simple or complex data sets. For the simple data set, outlier detection techniques are divided into parametric, semi-parametric and non-parametric methods. Distribution-based, depth-based and graph-based techniques are proposed for parametric approaches. Clustering-based, distance-based and density-based techniques are proposed for non-parametric approaches. Neural network-based and support vector machine-based techniques are proposed for semi-parametric approaches. On the other hand, for the complex data sets, outlier detection techniques are grouped depending on different types of data sets described in Section 2. Specifically, subspace-based and distance-based techniques are proposed for high dimensional data sets. Graph-based techniques are proposed for mixed-type attributes data sets. Clustering-based and tree-based techniques are proposed for sequence data sets. Graph-based and distribution-based techniques are proposed for spatial data sets. Model-based, graph-based and density-based techniques are proposed for streaming data sets and clustering-based and distribution-based techniques are proposed for spatial-temporal data sets. Supervised and Semi-supervised approaches can also employ some outlier detection techniques addressed in unsupervised approaches, although they initially need to train on pre-labelled data. This topic is outside the scope of this paper.

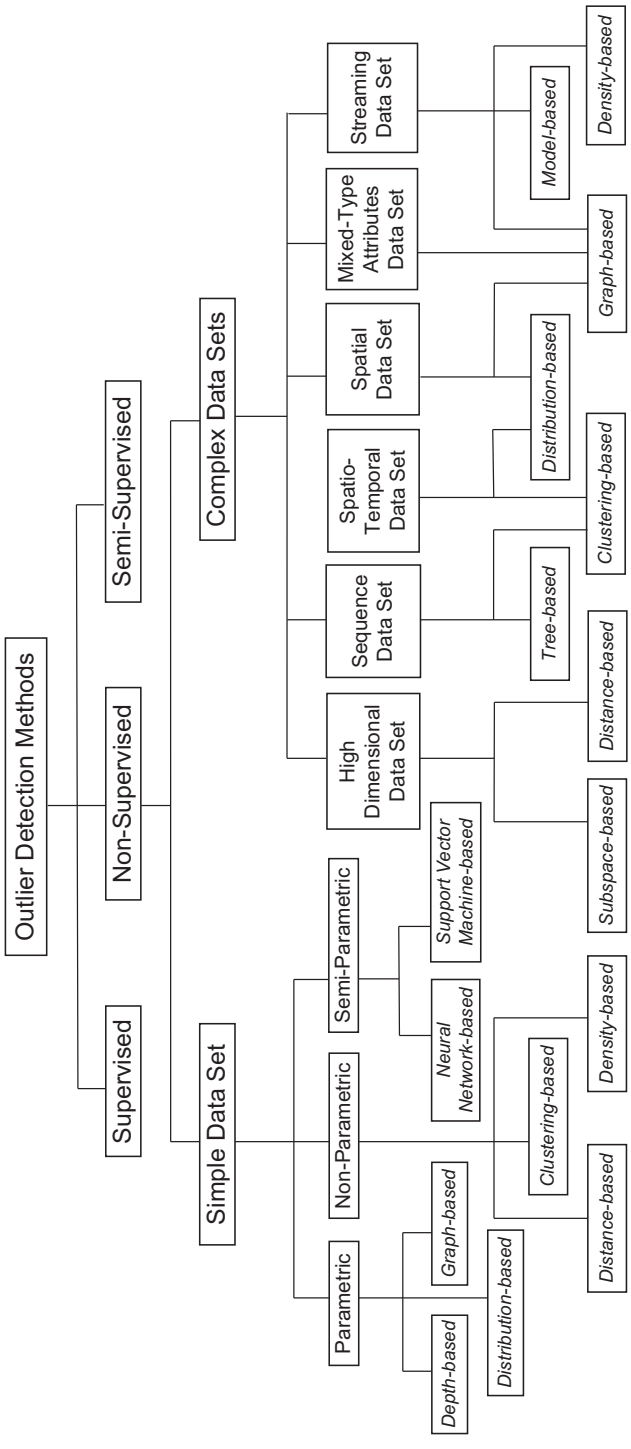

Fig. 2. Taxonomy of outlier detection techniques

3.2 Overview of Outlier Detection Methods

Early work in outlier detection was done in the field of *statistics*. *Distribution-based* methods assume that the whole data follow a *standard statistical* distribution model and determine a point as an outlier depending on whether the point deviates significantly from the data model. These methods can fast and effectively identify outliers on the basis of an appropriate probabilistic data model. *Depth-based* methods use the concept of computational geometry and organize data points in layers in multi-dimensional data spaces. Each data point is assigned a *depth* and outliers are those points in the shallow layers with smaller depth values. These methods avoid the problem of fitting into data distribution. *Graph-based* methods make use of a powerful tool *data image* and map the data into a *graph* to visualize the single or multi-dimensional data spaces. Outliers are those points that are present in particular positions of the graph. These methods are suitable to identify outliers in real-valued and categorical data.

Outlier detection also attracts much attention from the *data mining* community. Traditional *clustering-based* methods are developed to optimize the process of *clustering* of data, where outlier detection are only by-products of no interest. The novel clustering-based outlier detection methods can effectively identify outliers as points that do not belong to clusters of a data set or as clusters that are significantly smaller than other clusters. *Distance-based* methods are used to identify outliers based on the measure of full dimensional *distance* between a point and its nearest neighbors in a data set. Outliers are points that are distant from the neighbors in the data set. These methods do not make any assumptions about the data distribution and have better computational efficiency than depth-based methods, especially in large data sets. *Density-based* methods are proposed to take the local *density* into account when searching for outliers. These methods can effectively identify local outliers in data sets with diverse clusters.

In addition, some classification techniques have been applied to outlier detection. *Unsupervised neural networks based* methods can autonomously model the underlying data distribution and distinguish between the normal and abnormal classes. Those data points that are not *reproduced* well at the output layer are considered as outliers. These methods effectively identify outliers and automatically reduce the input features based on the key attributes. *Unsupervised support vector machine based* methods can distinguish between the normal and abnormal classes by mapping data into the feature space. Those points that are distant from most other points or are in relatively sparse regions of the *feature space* are declared as outliers. These methods effectively identify outliers without pre-labelled data.

Being concerned with the complex data sets, several novel outlier detection methods have been proposed to deal with data with specific semantic. *Subspace-based* methods project the data into a low-dimensional *subspace* and declare a point as an outlier if this point lies in an abnormal lower-dimensional projection, where the density of the data is exceptionally lower than the average. These methods reduce the dimensions of data and efficiently identify outliers in high dimensional data sets. *Tree-based* methods construct a specific *tree* as index to decompose data structure and use an efficient similarity measure for the sequence data to distinguish outliers from non-outliers. These methods efficiently identify outliers only

by examining nodes near the root of tree. *Model-based* methods detect outliers by the construction of a *model*, which can represent the statistical behavior of data stream. Outliers are those points that deviate significantly from the learned model. These methods can efficiently deal with the streaming data in an online fashion.

3.3 Prevalent Definitions of Outlier

The definitions of an outlier have been differently introduced by various outlier detection techniques. Being defined differently, they all aim at identifying instances of unusual behavior when compared to the rest majority of observations. As shown in Table I, we present a collection of prevalent definitions of outliers with respect to specific method-based outlier detection techniques. Clearly, there is no universally accepted definition exists.

Table I. Prevalent definitions of outliers.

Author	Definition
Hawkins [4]	*Distribution-based outlier*: An outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.
Barnett and Lewis [5]	*Distribution-based outlier*: An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.
Rousseeuw and Leroy [19]	*Distribution-based outlier*: Let T be observations from a univariate Normal distribution $N(\mu, \sigma)$ and o is a point from T. Then the Z-score for o is greater than a pre-selected threshold iff o is an outlier.
Rousseeuw and Leroy [19]	*Depth-based outlier*: Depth-based outliers are points in the shallow convex hull layers with the lowest depth.
Laurikkala et al. [29]	*Graph-based outlier*: Outliers are points that are present in particular positions of the graph.
Yu et al. [38], Jiang et al. [39]	*Clustering-based outlier*: Outliers are points that do not belong to clusters of a data set or as clusters that are significantly smaller than other clusters.
Knorr and Ng [42]	(i) *DB(f,D) outlier*: An object o in a data set T is an outlier if at least a fraction f of the objects in T lies at a greater distance than D from o. (ii) *DB(k,D) outlier*: An object o in a data set T is an outlier if at most k objects in T lie at distance at most D from o.
Ramawamay et al. [44]	DB_n^k *outlier*: The top n points with the maximum distance to their own k^{th} nearest neighbor are considered as outliers.
Angiulli and Pizzuti [70]	DB_ω^k *outlier*: Given an integer k, the weight ω of a point is defined as the sum or average of the distances separating it from its k nearest-neighbors. Outliers are those points scoring the largest values of weight.
Breunig et al. [46]	*Density-based outlier*: Outliers are points that lie in the lower local density with respect to the density of its local neighborhood.
Hu and Sung [50]	*Density-based outlier*: A point can be considered as an outlier if its own density is relatively lower than its nearby high density pattern cluster, or its own density is relatively higher than its nearby low density pattern regularity.
Hawkins et al. [57]	*Neural network based outlier*: Points that are not reproduced well at the output layer with high reconstruction error are considered as outliers.
Scholkopf et al. [61]	*Support vector machine based outlier*: Points that are distant from most other points or are present in relatively sparse regions of the feature space are considered as outliers.
Aggarwal and Yu [64]	*Subspace-based outlier*: A point is considered to be an outlier if in some lower-dimensional projection it is present in a local region of abnormal low density.
Muthukrishnan et al. [93]	*Time series streaming outlier*: If the removal of a point from the time sequence results in a sequence that can be represented more briefly than the original one, then the point is an outlier.
Shekhar et al. [82]	*Spatial outlier*: A spatial outlier is a spatially referenced point whose non-spatial attribute values are significantly different from those of other spatially referenced points in its spatial neighborhood.
Cheng and Li [98]	*Spatio-temporal outlier*: A spatial-temporal point whose non-spatial attribute values are significantly different from those of other spatially and temporally referenced points in its spatial or/and temporal neighborhoods is considered as a spatial-temporal outlier.

4. OUTLIER DETECTION TECHNIQUES FOR SIMPLE DATA SET

In this section, we describe method-based outlier detection techniques for the simple data set. Specifically, we summarize main ideas and relevant features of these techniques, and also give a brief evaluation for each outlier detection category.

4.1 Distribution-Based Method

Distribution-based methods, as typical *parametric* methods, are the earliest approach to deal with the outlier detection problem. They assume that the whole data follow a statistical distribution (e.g., Normal, Poisson, Binomial) and make use of mathematics knowledge of applied statistics and probability theory to construct a data model. They employ statistical tests to determine a point as an outlier depending on whether it deviates significantly from the data model.

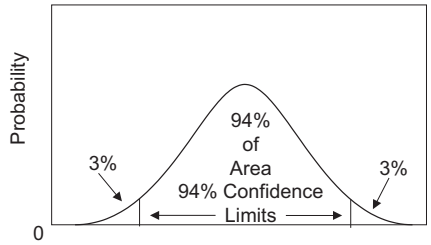

Fig. 3. An example of distribution of points

Grubbs and Frank [18] initially carry out the test on detecting outliers in a *univariate* data set. They assume that the whole data follows a standard statistical *t-distribution* and aim to identify *one* outlier at each iteration. In particular, each point has it own G value, which is the Grubbes test statistic value and can be calculated based on the sample mean and standard deviation. If the G value of a point is greater than a threshold value, i.e., the upper critical value of the t-distribution, the hypothesis of no outliers is rejected at corresponding significance level. Furthermore, the point is identified as an outlier and immediately eliminated from the data set. The procedure will be repeated until no more outliers are detected. The approach does not require user-defined parameters and all parameters are calculated directly from original data. However, multiple iteration may change the probabilities of detection and influence the accuracy of the test.

Three most important fundamental textbooks [4, 5, 19] concerning with outlier detection present classical definitions of distribution-based outliers respectively, as shown in table (0**??**). Also, Barnett and Lewis [5] and Rousseeuw and Leroy [19] further address a comprehensive description and analysis of *statistical* outlier detection techniques. They discuss the problem of detecting outliers in *univariate* and *multivariate* data. In detecting univariate outliers, they assume that data points can be modelled by a statistical standard distribution, usually the Gaussian (normal) distribution being used. The statistical distribution includes two parameters, the mean and standard deviation. Based on the observation that the probability of a point that lies outside *three* standard deviations away from the mean is only

0.0027, thus, three standard deviations is used as a threshold to determine how significantly a point deviates from the data model, as shown in Figure 3. Alternatively, a simplified *Z-score* function that more directly represents the degree of anomaly of each point is defined as:

$$Z = (x - \mu)/\sigma \tag{1}$$

Where μ is the mean, σ is the standard deviation. If the absolute value of Z-score of a data point is greater than 3, the point is declared as an outlier. On the other hand, in detecting multivariate outliers, they usually assume a multivariate normal distribution to represent the data. In order to use a simple threshold to determine whether a point is an outlier or not, the *Mahalanobis distance*, an effective distance measure, can take the shape of the multivariate data distribution into account and identify the attribute correlations accurately. For a d-dimensional multivariate sample x_i $(i = 1; n)$, the Mahalanobis distance is defined as:

$$MD_i = \sqrt{(x_i - t)^T \Sigma^{-1} (x_i - t)} \tag{2}$$

where Σ represents the $d \times d$ covariance matrix and t is the multivariate mean. Furthermore, for N d-dimensional points from a normal distribution, the square of Mahalanobis distance follow a chi-square distribution (χ^2_d) with d degree of freedom [20]. Thus, an outlier in multivariate data is a point whose Mahalaobis distance is larger than a pre-defined threshold. Alternatively, Euclidean distance is another basic distance measure and is defined as:

$$ED_i = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

where x_i, y_i are two points and n is the dimensionality of the data. The Euclidean measure is seldom used in distribution-based outlier detection approaches since it cannot effectively capture the shape of the multivariate data distribution. The authors further carry out numerous discordance tests under different circumstances depending on the data distribution, the distribution parameters, the number of expected outlier and the types of expected outlier. The testing results show that the approach achieves good performance in finding outliers in univariate data.

Based on their work, [21, 22] propose robust outlier detection approache based on the minimum covariance determinant (MCD), which aims at alleviating the problem that the mean and standard deviation of the distribution may be extremely sensitive to outliers during the computation of Mahalanobis distance. The main idea of MCD is to only use a *subset* of points, which are the minimum number of non-outliers to minimize the determination of the covariance matrix.

Eskin [23] proposes a *mixture model* approach to detect outliers in *univariate* data. The author assumes that the data is modelled as a mixture of two distributions M and A, which represent the majority of normal points and the minority of anomalous points respectively. Each point in the data set is fallen into either M or A based on corresponding probability value λ. Initially, all of data points are put in the set of M while the set of A is empty. The probability function of the entire data may change by moving a data point from M to A. If the difference of the probability function value is larger than a pre-defined threshold c, the point is

declared as an outlier and then moved permanently to A. The procedure will be repeated until every point in the set of M experiences the comparison. The choice of two user-defined parameters λ and c is very important and may greatly influence the performance of this approach.

Yamanishi et al. [24] present a *mixed model* of Gaussian distribution to represent the normal behaviors of data in the detection of outliers. Each data point is assigned a score based on the degree to which the point deviates from the model. A higher score of a point indicates that the point is more likely to be an outlier. This outlier detection approach can be used to handle categorical and continuous variables.

4.1.1 *Evaluation of Distributed-based Techniques.* Distribution-based approaches are mathematically justified and can effectively identify outliers if a correct probabilistic data model is given. Also, the construction of the data model helps to store minimal amount of information to represent the model, instead of the entire actual data. However, distribution-based techniques suffer from two serious problems. Firstly, these techniques only work well in a single-dimensional data set so that they are not suitable to identify outliers in even moderately high dimensional spaces. Secondly, in many real-life situations, a priori knowledge of data distribution is not available. Finding a possible standard distribution that fits the data is computationally expensive and eventually may not produce satisfactory results.

4.2 Depth-Based Method

Depth-based methods exploit the concept of computational geometry [25] and organize data points into layers in k-dimensional data space. Based on the definition of *half-space depth* [26], also called as *depth contours*, each data point is assigned a *depth* and outliers are those points in the shallow layers with smaller depth value.

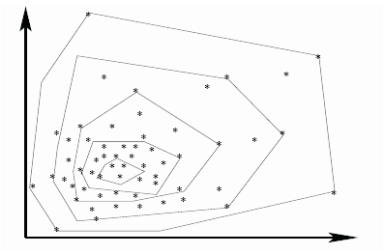

Fig. 4. An example of depth of points [26]

Rousseeuw and Leroy [19] describe two basic depth-based outlier detection techniques for low dimensional data sets, i.e., *minimum volume ellipsoid* (MVE) and *convex peeling*. MVE uses the smallest permissible ellipsoid volume to define a boundary around the majority of data. Those points are outliers if they are not in the densely populated normal boundary. Convex peeling maps data points into convex hull layers in data space according to peeling depth. Outliers are those points in the shallow convex hull layers with the lowest depth. Both MVE and convex peeling are robust outlier detection techniques that use the specific percentages of data points to define the boundary. Thus, these outlying points will not skew their

boundary. The key difference between the two techniques is how many outliers are identified at a time. In particular, MVE maintains all data points to define a normal boundary, then removes *multiple* outliers at once, while convex peeling builds convex hull layers and then peels away *one* outlier with the lowest depth at a time. The procedure will be repeated until a pre-defined number of outliers have been removed from the data set.

Based on [19], Ruts and Rousseeuw [27] present an outlier detection approach using the concept of *depth contour* to compute the depth of points in a *two*-dimensional data set. The deeper the contour a data point fits in, the more robust it is regarded as an outlier. Johnson et al. [28] further extend the work of [27], and propose a *faster* outlier detection approach based on computing *two*-dimensional depth contours in convex hull layers. In particular, this approach only needs to compute the first k depth contours of a selected subset of points, instead of the entire data as it is done in [27] and it is robust against collinear points.

4.2.1 *Evaluation of Depth-based Techniques.* Depth-based approaches avoid the problem of fitting to a data distribution but instead compute multi-dimensional convex hulls. However, they are inefficient for the large data set with high dimensionality, where the convex hull will be harder to discern and is computationally more expensive. Experimental results show that existing depth-based methods provide acceptable performance for only up to 2-dimensional space.

4.3 Graph-Based Method

Graph-based methods make use of a powerful tool *data image*, i.e., map the data into a graph to visualize the single or multi-dimensional data spaces. Outliers are expected to those points that are present in particular positions of the graph.

Laurikkala et al. [29] propose an outlier detection approach for univariate data based on *box plot*, which is a simple *single*-dimensional graphical representation and includes five number values: lower threshold, low quartile, median, upper quartile and upper threshold. Figure 5 shows an example of a box plot. Using box plot, points that lie outside the lower and upper threshold are identified as outliers. Also, these detected outliers can be ranked by the occurrence frequencies of outliers. Thus, the box plot effectively identifies the top n outliers with the highest occurrence frequencies and then discards these outliers. The approach is applicable for real-valued, ordinal and categorical data. However, it is too subjective due to excessively rely on experts to determine several specific points plotted in the graph, e.g., low and upper quartile.

Scatter plot [30] is a graphical technique to detect outliers in *two*-dimensional data sets. It reveals a basic linear relationship between the axis X and Y for most of the data. An outlier is defined as a data point that deviates significantly from a linear model. Figure 6 shows an example of a scatter plot. In addition, spin plot [31] can be used for detecting outliers in *3*-D data sets. D-D plot [22] is used to illustrate the relationship between a MCD-based robust distances [21] and the full Mahalanobis distances on detecting outliers. Marchette and Solka [32] use *interpoint distance measure* to order the data image in data sets to roughly group outliers according to how anomalous they are.

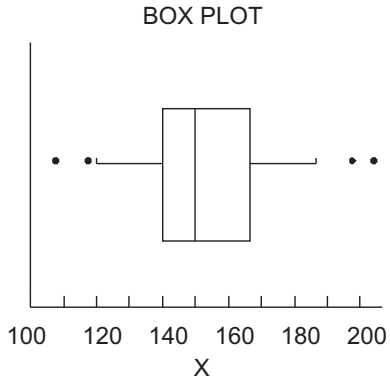

Fig. 5. An example of a box plot

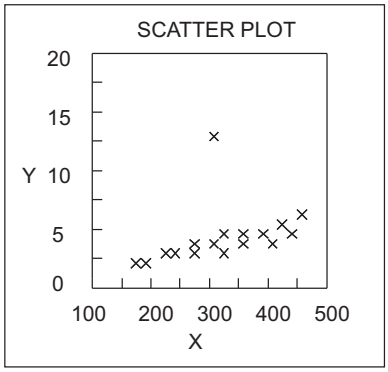

Fig. 6. An example of a scatter plot.

4.3.1 *Evaluation of Graph-based Techniques.* Graph-based approaches have no assumptions about the data distribution and instead exploit the graphical representation to visually highlight the outlying points. They are suitable for identifying outliers in real-valued and categorical data. However, they are limited by the lack of precise criteria to detect outliers. In particular, several specific points in the graph are determined subjectively by experts, which is also a very time-consuming and difficult process.

4.4 Clustering-Based Method

Clustering-base methods use one data mining technique, i.e., *clustering*, to effectively and efficiently cluster data. Traditional clustering-based approaches, e.g., DBSCAN [33], CHAMELEON [34], BIRCH [35], CURE [36] and TURN [37] are developed to optimize the process of clustering rather than detect outliers. Also, they do not have a formal and acceptive definition for outliers. Thus, detecting outliers in these approaches are only by-products of no interest. Here, we describe several novel outlier detection approaches, which are designed specially for detecting outliers based on clustering techniques. In these approaches, outliers are points that do not belong to clusters of a data set [38] or are clusters that are significantly

smaller than other clusters [39]. As shown in Figure 7, the points O_1, O_2 and the cluster C_1 are outliers. The detection of local outliers in clusters is addressed in [40, 41] .

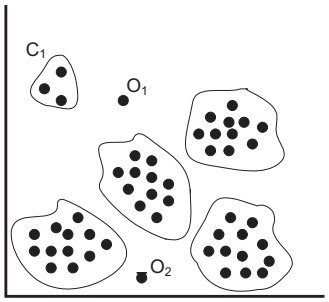

Fig. 7. An example of clusters of points

Yu et al. [38] propose an outlier detection approach based on a signal-processing technique *wavelet transform*, which has the multi-resolution property and can be extended to detect outliers in data sets with different densities. In particular, this approach uses wavelet transform to quantize the data space and finds the dense clusters in the transformed space. By removing clusters from the original data, the remaining points in the non-dense clusters are labelled as outliers. This approach cannot measure the degree of outlierness, and also do not detect outliers by considering the distance between small clusters and their closest large cluster.

Jiang et al. [39] present a *two-phase* clustering approach to identify outliers. In the first phase, this approach partitions the data into clusters based on a *heuristic* instead of the traditional k-means algorithm, which is optimized to search for a fixed number of clusters. The used heuristic states "*if the points in the same cluster are not close enough, the cluster can be split to two smaller clusters*". This helps to reduce the time complexity since the data is processed as fractions and not as whole. In the second phase, this approach employs an outlier-finding process (OFP) to identify outliers based on the construction of a minimum spanning tree (MST), which can remove the longest edge of the tree. Eventually, the small clusters with less number of nodes in the tree are considered as outliers.

He et al. [40] introduce a new definition of a cluster-based *local* outlier, which takes both the size of a point's cluster and the distance between the point and its closest cluster into account. Each point is associated with a cluster-based local outlier factor (CBLOF), which is used to determine the degree of its outlierness. The proposed approach first partitions the data into clusters by using a *squeezer* algorithm, which only makes one scan over the data set and produces good clustering results. Then the outlier factor is computed for each point and outliers are points which have the largest values. This approach has the linear scalability with respect to the size of data and can work well in large data sets.

Ren et al. [41] propose a more efficient clustering-based *local* outlier detection approach by combining the detection of outliers with grouping data into clusters in a *one-time* process. This approach does not require the beforehand clustering

process addressed in [38, 39, 40]. The degree of a point's outlierness is measured by a local connective factor (LCF), which indicates how significantly the point connects with other points in the data set. Specifically, LCF is further calculated by a vertical data representation *P-Tree* [41], which uses logical operations to efficiently compress the data and can be used as an index for the pruning. Outliers are those points that are not connected with clusters. Experimental results show that this approach has better performance in terms of efficiency for large data sets compared to approaches reported in [39, 40].

4.4.1 *Evaluation of Clustering-based Techniques.* Clustering-based approaches do not require a priori knowledge of data distribution and exploit clustering techniques to efficiently filter and remove outliers in large data sets. In particular, novel clustering-based approaches have been developed to optimize the outlier detection process and reduce the time complexity with respect to the size of data. However, these approaches are suspectable to high dimensional data sets since they rely on the full-dimensional distance measure of points in clusters.

4.5 Distance-Based Method

Distance-based methods, as typical *non-parametric* methods, identify outliers based on the measure of full dimensional distance between a point and its nearest neighbor in the data set. *Euclidean distance* is commonly used as a similarity measure in distance-based methods.

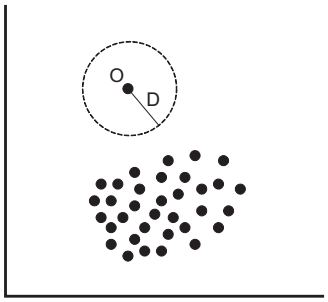

Fig. 8. An example of a distance of an outlier

Knorr and Ng [42, 43] define a distance-based outlier as, "*a point* o *in a data set* T *is a* DB(p,D) *outlier if at least a fraction* p *of the points in* T *lies at a greater distance than* D *from* o". An example of a distance-based outlier is hown in Figure 8. Based on this definition, the detection of outliers relies on Euclidean distance to measure similarity between every pair of points. The two parameters D and p are used to determine the range of neighborhood. The authors further propose *three* outlier detection algorithms, i.e., index-based, nested-loop and cell-based. The *index-based* algorithm is based on a priori constructed index structure and executes a range search with radius D for each point. If more than $M = (1-p)N$ neighbors are found in a point's D-neighborhood, the search will stop and the point is declared as a non-outlier, otherwise it is an outlier. The *nested-loop* algorithm avoids the

cost of preliminary construction of the index, and instead partitions the entire set of points into blocks and then directly computes the distance between each pair of points in the blocks. A point that has less than M neighbors within the distance D is declared as an outlier. The two algorithms have the same time complexity of $O(k\ N^2)$, where k is the dimensionality and N is the number of points in the data set. The *cell-based* algorithm partitions the entire data set into cells and effectively prunes away a large number of non-outlier cells before finding out outliers. This helps to speed up outlier detection by only detecting outliers in the subset of cells, which may include potential outliers. These algorithms depend on two user-defined parameters D and p, which are not usually easy to determine. Also, they do not provide a ranking on the degree of outliers.

Ramaswamy et al. [44] further extend the outlier definition in [42] based on the distance of a point from its k^{th} nearest neighbor, instead of the estimation of an appropriate distance D. Also, they provide a ranking of the top n outliers by the measure of the outlierness of points. Their novel definition of distance-based outliers is that the top n points with the maximum distance to their own k^{th} nearest neighbor are considered as outliers. The authors also exploit the index-based and nested-loop algorithms to detect outliers. Furthermore, they propose a *partition-based* algorithm to prune a significant number of partitions and efficiently identify the top n outliers in the rest of partitions from the data. Experimental results show that this partition-based algorithm reduces the cost of computation and I/O in large and multi-dimensional data sets. However, these algorithms suffer from the choice of the input parameter k. Also they only consider the distance to the k^{th} nearest neighbor and ignore distance to other closer points.

Bay and Schwabacher [45] propose an optimized nested-loops algorithm that has *near linear* time complexity on mining the top n distance-based outliers. They randomize the data and partition the data into multiple blocks. Each point is associated with an anomaly score, which is determined by either the distance to its k^{th} nearest neighbor or the average distance to its k nearest neighbors. The top n distance-based outliers with the largest scores initially are identified in the first block. Then the smallest score of them is used as a *cut-off* for the rest of blocks. If a point in other blocks has a larger score than the cut-off, the cut-off will be increased and replaced by the smallest score in the new n outliers, otherwise, the point will be pruned. Eventually, n extreme outliers are identified and the cutoff increases along with pruning efficiency. However, the algorithm is sensitive to the order of data and the distribution of the data set. If the data is sorted or correlated, the performance is poor and time complexity is of quadratic order in the worst case.

4.5.1 *Evaluation of Distance-based Techniques.* Distance-based approaches do not make any assumption about the data distribution and are computationally more efficient than the depth-based approaches for large data sets. The proposed distance-based outliers definitions can generalize many notions from distribution-based approaches. However, they rely on the existence of some well-defined notions of distance to measure the similarity between two data points in the entire space, which is not easy to define in high dimensional data sets. Also, they only identify outliers in a global view and are not flexible to discover local outliers, especially in data sets which have diverse densities and arbitrary shapes.

4.6 Density-Based Method

A key problem of distance-based approaches is that they suffer from detecting local outliers in a data set with diverse densities. For example, as shown in Figure 9, two points O_1 and O_2 are viewed as outliers with respect to the clusters C_1 and C_2, respectively. However, O_2 may not be an outlier using distance-based methods due to the fact that C_2 is too dense relative to C_1. Thus, density-based approaches are proposed to solve this problem by taking *local density* into account when searching for outliers. The computation of density still depends on full dimensional distances measure between a point and its nearest neighbors in the data set.

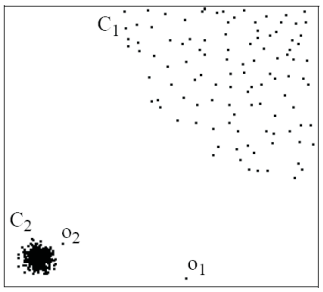

Fig. 9. A problem of distance-based methods in a data set with different densities [46]

Breunig et al. [46] originally introduce the notion of density-based local outliers based on the density in the local neighborhood. Each data point is assigned a local outlier factor (LOF) value, which is calculated by the ratio of the local density of this point and the local density of its *MinPts* nearest neighbors. The single parameter *MinPts* of a point determines the number of its nearest neighbors in the local neighborhood. The LOF value indicates the degree of being an outlier depending on how isolated the point is with respect to the density of its local neighborhood. Points that have the largest LOF values are considered as outliers.

Based on the work reported in [46], many novel density-based approaches [47, 48, 49, 50, 51, 52, 53] have been developed to further improve the effectiveness and efficiency of LOF. Also, [54] and [55] combine distance-based and density-based approaches to identify outliers in a data set.

Chiu and Fu [47] present *three* enhancement schemes for LOF called $LOF^{'}$, $LOF^{''}$ and *GridLOF*. The first two schemes are variants of the original LOF computation formulation. $LOF^{'}$ provides a simpler LOF formulation by replacing *local reachability density* with *MinPts-dist*, where local reachability density indicates the local density of a point's *MinPts* nearest neighbors, and *MinPts*-dist indicates the distance to the point's *MinPts* nearest neighbors. This scheme also reduces one scan over the data on the computation of local reachability density. $LOF^{''}$ is a generation of LOF by using two different *MinPts* values to define a point's neighborhood and the point's neighbors' neighborhood, and can capture local outliers within different circumstances. The third scheme *GridLOF* uses a simple *grid-based* technique to prune away some non-outliers and then only computes the LOF values for the remaining points. This helps to reduce the computation of LOF for all points,

however, the deletion of points may affect the LOF values of those points in their own neighborhood.

Jin et al. [48] propose an efficient outlier detection approach, which only determines the top n local outliers with the maximal LOF values and reduces the computation load of LOF in [46] for all points. The approach does not find LOF values for all points to select the top n outliers, and does not also perform the process of immediately pruning non-outliers before detecting potential outliers. Instead, it first uses an efficient clustering technique, i.e., BIRCH [35] to compress the data into micro-clusters, where a group of data is so close together, and then computes LOF upper and lower bounds for each micro-cluster. The micro-clusters with the largest LOF low bound are chosen to identify the top n local outliers. The approach suffers from the choice of the parameter *MinPts*.

Tang et al. [49] present an outlier detection approach more effective than [46], especially for *sparse* data sets, where the non-outlier patterns may have low densities. The approach uses a connectivity-based outlier factor (COF) value to measure the degree of outlierness and takes both the density of the point in its neighborhood and the degree that the point is connected to other points into account. The COF can be calculated using the ratio of the average distance from the point to its k-distance neighbors and the average distance from its k-distance neighbors to their own k-distance neighbors. Points that have the largest COF values are declared as outliers.

Hu and Sung [50] consider the problem of detecting outliers in a data set, where *two* patterns exist, i.e., high density clustering and low density regularity. The latter pattern consists of a set of regularly spaced points whose density are lower than that of their neighboring outliers. The authors introduce a new definition of an outlier according to these two different patterns, "*if a point's own density is relatively lower than its nearby high density pattern cluster, or its own density is higher than its nearby low density pattern regularity, the point can be declared as an outlier*". The definition enhances the effectiveness of LOF, which does not work well in the low density regularity. The proposed approach uses a variance of volume (VOV) value to measure the degree of being an outlier and has similar time complexity with LOF. Points that have the largest VOV values are declared as outliers. This approach depends on a choice of a parameter constant intensity, which is used to decide the density of clusters.

Papadimitriou et al. [51] present a fast outlier detection approach called LOCI to detect local outliers based on the concept of a multi-granularity deviation factor (MDEF) value, which is used to measure a point's relative deviation of its local neighborhood density from the average local neighborhood density in its neighborhood. To alleviate the difficulty of choosing values for *MinPts* in [46, 47], LOCI uses a different definition for the local neighborhood, where each point has the same radius, instead of the fixed number of neighbors. A point can be declared as an outlier by comparing its MDEF with a derived statistical value, which is automatically derived from the data. Experimental results show that LOCI achieves good performance to accurately identify outliers without the user-defined threshold. However, the choice of an appropriate user-defined radius of the local neighborhood becomes a critical issue, especially for high dimensional data sets.

Kim et al. [52] propose an outlier detection approach, which uses the distance between a data point and its closest *prototypes* as the degree of outlierness. Prototypes refer to a small percentage of representative data from the original data can be identified by using the k-means technique. Then the outlierness of these prototypes are calculated by taking the measure of distance and density into account. Prototypes with the largest values will be removed. Finally, the approach measures the degree of outlierness of each original data point depending on its distance to its closest remaining prototypes. Outliers are points that are far from their prototypes with the largest distance values. Prototypes are not easily accurately determined, especially in data sets with different densities.

Ren et al. [53] develop an efficient density-based outlier detection approach based on a relative density factor (RDF) value, which is a local density measurement to measure the degree of being an outlier by contrasting the density between a point and its neighbors. The approach uses *P-Trees* to efficiently prune some non-outliers, and then only computes the RDF value for the remaining small subset of the data. Outliers are points whose RDF values are greater than a pre-defined threshold. Experimental results show that the approach has better performance than LOF and LOCI in terms of efficiency for large data sets.

Fan et al. [54] introduce a novel outlier notion by considering both *local* and *global* features of the data set. They define an outlier as a point that is inconsistent with the majority of the data or inconsistent with a group of it neighbors. The proposed approach uses a clustering technique, i.e., TURN [26] to efficiently identify outliers by consecutively changing the resolution of a set of data points. The resolution-based outlier factor (ROF) is used to measure the degree of outlierness of a point. Outliers are the top n points with the lowest ROF values. The approach does not need any input parameters and is more effective to detect outliers than DB-outlier and LOF.

Kollios et al. [55] propose a density-based biased sampling approach to detect DB-outlier based on *kernel density estimator*, which makes use of some randomly sampled points to represent the density of the whole data set and efficiently approximate the underlying probability distribution. Outlier detection can be performed based on measuring how many points are present within the distance D from a data point in the data set. After one single-pass through the data, those points that have less number of neighbors in their own neighborhood than a specified threshold are considered as outliers. The performance of this approach depends on the accuracy of the density estimator.

4.6.1 *Evaluation of Density-based Techniques.* Density-base approaches have no assumption about the data distribution and effectively identify local outliers in data sets with diverse clusters. However, a weakness of density-based approaches is the determination of input parameters, which are usually based on a priori or trail-and-error estimation. Also, the density definition is based on full dimensional distance computation between points, which is susceptible for high dimensional data sets.

4.7 Neural Network Based Method

Neural networks (NN) are often used in safety-critical applications for regression or classification purpose [56]. They can autonomously model the underlying data

distribution and distinguish the normal/abnormal classes. Recently, unsupervised neural network methods have been applied for outlier detection [56, 57, 58, 59]. Neural networks do not require pre-labelled data to permit learning and can identify those data points that are not reproduced well at the output layer as outliers. The *reconstruction error* can be used as the measure of outlierness for data points.

Sykacek [56] presents an outlier detection approach using the *equivalent error bar* [56] to identify outliers in the trained network with multi-layer perception. Outliers are points that are residual outside the equivalent error bar depending on a pre-defined threshold.

Hawkins et al. [57, 58] present an outlier detection approach for large multivariate data sets based on the construction of replicator neural networks (RNN), which is a variant of the usual regression model. Specifically, RNN is a feed-forward multi-layer perception neural network and contains three hidden layers between the input and output layers. Figure 10 shows a schematic view of the fully connected RNN. The use of RNN is to reproduce the input points at the output layer with minimized reconstruction error. If some small number of input points are not reconstructed well and cause high reconstruction errors in the trained neural network, these points can be considered as outliers. An outlier factor is used to measure the degree of outlierness of each point based on the average reconstruction error. Experimental results show that this method is effective for detecting network intrusion.

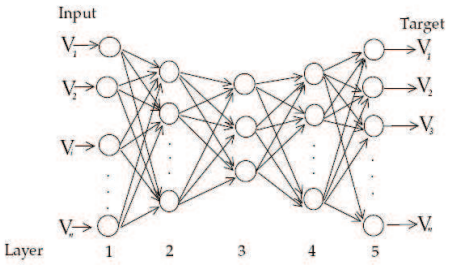

Fig. 10. Structure of a replicator neural network [57]

Fu and Yu [59] propose an outlier detection approach based on artificial neural network (ANN), which can be trained by different training algorithms. The authors describe three different structures of the trained ANN, i.e., one has no hidden layer, the other two ones have a hidden layer. Outlier detection can be performed in the three ANNs by a modified *Z-score*, which is used to measure the degree of outlierness of data points. If points are statistically inconsistent with the trained neural network, they are declared as outliers.

4.7.1 *Evaluation of Neural Network based Techniques.* Neural network based approaches, strictly belong to a *semi-parametric* method and are trained to model the underlying data distribution without a priori assumption on the properties of data. They are used to effectively identify outliers and automatically reduce the input features based on the key attributes. However, they are susceptible to

high dimensional data sets, where neural networks are harder to be trained well. Moreover, they need extra training time and are also sensitive to model parameters.

4.8 Support Vector Machine Based Method

Support vector machine (SVM) based methods are often used to deal with classification or regression problems by the data mining community [61]. They can distinguish between normal and abnormal classes by mapping data into the feature space. Recently, unsupervised SVM-based methods have been used for outlier detection [61, 62, 63]. They do not require a pre-labelled data set to determine a boundary region and can efficiently identify outliers by using the *kernel functions*.

Scholkopf et al. [61] present an outlier detection approach, which uses a kernel function (radial basis function) to efficiently map the original data into a vector space typically of high dimensions (feature space). Outlier detection can be performed depending on the position of points in the feature space. Those points that are distant from most other points or are in relatively sparse regions of the feature space are labelled as outliers. However, the performance of the approach is quite susceptible to the choice of variance parameter of radial basis function, which determines the size of the boundary regions.

Tax and Duin [62] further propose an outlier detection approach based on unsupervised SVM, called support vector domain description (SVDD). The approach uses a Gaussian kernel function to map the whole data set to high dimensional feature space. Then SVDD can classify normal data into one class from the rest of the feature space by learning an optimal hypersphere, which is a sphere with minimum volume, containing the majority of data points. Those points that belong to the boundary of the sphere are called support vector. On the contrary, the points that lie outside this sphere are considered as outliers. The approach does not measure the degree of being an outlier for each data point and it only determiners if a point is an outlier or not based on discrete outlier factor.

Petrovskiy [63] exploits the same idea of SVDD and presents an outlier detection approach by using kernel functions and fuzzy set theory. Instead of learning a sphere in the feature space, the approach constructs one fuzzy cluster, which contains all points from the original space, and then uses a continuous decision function to measure the degree of outlierness of each point in the cluster. Those points whose outlier factor is less than a threshold are considered as outliers. This approach provides the degree of outlierness.

4.8.1 *Evaluation of Support Vector Machine based Techniques.* Unsupervised SVM-based approaches do not have any assumption on data distribution and can effectively identify outliers without pre-labelled data. However, the computation of the kernel functions is a computationally expensive task. Also, it is not easy to determine appropriate parameters to control the size of boundary region.

5. OUTLIER DETECTION TECHNIQUES FOR COMPLEX DATA SETS

In this section, we describe outlier detection techniques designed specially for the complex data sets such as high dimensionality, mixed-type attributes, sequence data, spatial data, data streams and spatio-temporal data sets. Specifically, we summarize main ideas and relevant features of these techniques, and also give a

brief evaluation for corresponding outlier detection methods.

5.1 Outlier Detection Techniques for High Dimensional Data Set

The outlier detection techniques we have introduced so far consider the full dimensional attribute space for detecting outliers and have good performance for relatively low-dimensional data sets. However, they will be susceptible to the problem of *the curse of dimensionality* [16]. In particular, in high dimensional spaces, where the data is sparse, convex hull or MVE becomes harder to discern, the notion of proximity is less meaningful and distance computation is computationally more expensive. Also, high dimensionality increases time complexity and makes it more difficult to accurately approximate the distribution of underlying the data. Thus, several approaches have been proposed specially for detecting outliers in high dimensional data sets. They can generally be classified into *subspace-based* [64, 65, 66, 67] and *distance-based* [68, 69, 70, 71] methods.

Aggarwal and Yu [64, 65] propose a *subspace-based* outlier detection approach by observing the density distribution of *projections*, which refer to clusters in a low-dimensional subspace. This approach first projects the data into a low-dimensional subspace, and defines that an *abnormal* lower-dimensional projection is one in which the density of the data is exceptionally lower than the average. A point is eventually considered as an outlier if it is located in an abnormal lower-dimensional projection. Specifically, an *evolutionary search* algorithm is used to determine low-dimensional projections since it can efficiently find hidden combinations of dimensions in which data is sparse and has a faster processing time than the naive brute-force search algorithm. A *sparsity coefficient* value represents the density of a low-dimensional projection and can be used to measure the outlierness of projections. The smaller the sparsity coefficient value, the more abnormal the projection is. The points in the projections with the top n smallest sparsity coefficient values are identified as outliers. In this approach, the choice of an appropriate value for a pre-defined parameter, i.e., the dimensionality of subspaces is not easy.

Based on this work, Zhu et al. [66] present a robust outlier detection approach specially for high dimensional data sets by incorporating directly user-defined *example outliers*. As a result, points can be considered as outliers if they are in an extremely low-dimensional subspace, which also includes so many examples that it is outstanding significantly than any other subspaces. Similar to [64], the authors employ the evolutionary search algorithm to efficiently determine the low-dimensional subspaces. Also, a *fitness function* is used to measure the outlierness of subspaces.

Shyu et al. [67] propose an outlier detection approach based on principal components analysis (PCA), which is a robust estimator of the correlation matrix of normal patterns. This approach sequentially executes two functions of principal components to identify outliers. The first function uses the major principal components to detect extreme points with large values of variances and covariances depending on the subset of original attributes. The second function uses the minor principal components to further identify the rest of outliers, which have different correlation structures from normal patterns. This approach based on PCA does not have any assumption on the data distribution and can be applied in real-time applications by effectively reducing the dimensionality of the data space without losing any valuable information.

Angiulli and Pizzuti [68, 69] design an approach to efficiently find out the top n outliers in large and high-dimensional data sets based on *distance-based* methods. The authors introduce a new definition of outliers, that is, *those points with the largest values of ω_k, which is the sum of the distance to their k nearest neighbors.* Based on this definition, outliers can be identified by two phases, approximate and exact solutions. The *approximate* solution first determines an approximate subset, which includes k candidate outliers. The *exact* solution further determines the true outliers from these candidate outliers. A *space-filling curve* is used to linearize the data set. However, the time complexity of the approach is still of quadratic order with the number of dimensionality in the worst case.

Ghoting et al. [70] extend the work of [45] and present a fast *distance-based* outlier detection approach. In particular, this approach uses a *divisive hierarchical clustering* to effectively partition the data set into clusters in terms of the similarity of distance, and further efficiently identifies outliers relying on a novel *nested loop* algorithm, which aims at finding a data point's k *approximate nearest neighbors*, which are within a cutoff threshold, instead of its nearest neighbors used in [45]. Experimental results show that this approach has near-linear time performance with the increase of dimensionality.

Chaudhary et al. [71] introduce an outlier detection approach based on *k-d tree*, which is a space decomposition data structure. This approach uses k-d tree to efficiently partition the whole data set into groups so that all objects in each group have the similar outlierness. As a result, those points in a sparse region are considered as outliers. In order to generate regions with the uniform density, an improved k-d tree with *special cuts* is used to separate out the clusters from the sparse regions. Furthermore, a *smoothing factor* value is used to measure the sparseness of the region around points. This approach does not scale well with the number of dimensions.

5.1.1 *Evaluation of Techniques for High Dimensional Data Sets.* Subspace-based approaches identify outliers by finding lower-dimensional projections or selecting the key subset of attributes. They effectively overcome the effects of the curse of dimensionality for traditional outlier detection approaches. However, it is not easy to choose appropriate values for the pre-defined parameters of lower-dimensional projections, and it is also is difficult to decide which are key attributes. Distance-based approaches still are computationally expensive in high dimensional data sets and the time complexity is of quadratic order in the worst case.

5.2 Outlier Detection Techniques for Mixed-Type Attributes Data Set

Most non-parametric approaches use the notion of distance or density to measure the similarity between two data points in continuous (numeric) data spaces. Similarly, distribution-based and neural network based approaches identify outliers in numeric or ordinal data sets. However, they will be susceptible to many real-life applications, where the data contains both continuous and categorical attributes, the latter of which only include partial ordering values. This mixed-attribute data is very difficult to measure by traditional distance metric, and also may influence the performance of detecting outliers if it is only simply disregarded. Thus, several approaches have been designed specially for detecting outliers in categorical or

mixed-type attributes data sets based on *graph-based* [72, 73, 74, 75, 76] methods.

Otey et al. [72, 73] present an approach to effectively identify outliers in a mixed attribute space by taking into account the dependencies between continuous and categorical attributes. An *anomaly score* function is used to compute three different dependencies, i.e., between (i) categorical attributes, (ii) continuous attributes and (iii) continuous and categorical attributes. Outliers are points that deviate from these dependencies. In a categorical attribute space, two data points are considered *linked* if they have at least one common attribute-value pair. The number of attribute-value pairs in common indicates the strength of the associated link between these two points. A data point can be considered as an outlier if it has very few links or very weak links to other points. In a mixed attribute space, the dependence between the values with mixed continuous and categorial attributes is captured by incrementally maintenance of *covariance matrix*. A data point can be considered as an outlier if the number of its attribute-value pairs that are infrequent and its corresponding covariance are violated from the dependencies between the mixed attributes. This approach suffers from the problem of dimensionality. Also, the construction of covariance matrix implies an assumption that the whole data has the same distribution, which is not easily available in real-life applications.

Wei et al. [74] propose an efficient approach for detecting *local* outliers in categorical data. This approach uses a *hypergraph* model to precisely capture the distribution characteristics in a data subspace. A hypergraph is a generalized graph, consisting of a set of vertices and hyperedges. Each vertex corresponds to a data point in the data set, and each hyperedge contains a group of points that have a frequent attribute-value pairs. In this model, a point can be considered as an outlier if the deviation of the point on an outlying attribute is less than a pre-defined threshold. In particular, this approach alleviates the problem of the curse of dimensionality depending on selecting the most relevant subspaces. Also, it uses the connectivity property of points to efficiently deal with missing-value data.

He et al. [75] present a *local-search heuristic* approach to identify outliers in categorical data. On the basis of a heuristic observation that the removal of some points from a data set will result in a data set that is less discord, this approach uses an *entropy* function to measure the degree of disorder of the rest of data set. A point is declared as an outlier if the entropy value after exchanging its label with each of the pre-defined outliers is decreased. This procedure will be repeated until k outliers are identified. Experimental results show that this approach scales well with the size of data sets.

Yu et al. [76] propose an outlier detection approach for detecting *centric* local outliers in categorical/numerical data, instead of detecting local outliers around bounder in LOF [46]. The similarity among points can be measured by a *similarity graph*, which is a weighted connected undirected graph. A weight value for a pair of points specifies the similarity between of the two points. A point can be considered as an outlier if its similarity relationship with its neighbors is lower than the similarity relationships among its neighbors' neighborhood. The use of the similarity graph overcomes the disadvantage of the traditional similarity measure and can easily be applicable for categorical/ordinal as well as numerical data.

5.2.1 *Evaluation of Techniques for Mixed-Type Attributes Data Sets.* Graph-based approaches effectively visualize the similarity relationships between the mixed-attribute data points and highlight outliers according to the degree of similarity relationships. They can overcome the disadvantage of traditional similarity measure for categorical data. However, they may suffer from the curse of dimensionality, and also lack strictly precise criteria for determining outliers.

5.3 Outlier Detection Techniques for Sequence Data Set

In some applications, the data is naturally represented as a sequence of individual entities, such as symbols or letters. It becomes very difficult for traditional distance and density-based outlier detection techniques to define a standard notion of similarity for measuring the structural differences between sequences. Thus, existing outlier detection techniques for sequence data sets exploits *clustering-based* [77] and *tree-based* [78] methods.

Budalakoti et al. [77] introduce an outlier detection approach for high-dimensional symbol sequences. This proposed approach efficiently clusters the sequence data into groups and find out anomalous subsequences that deviates from normal behaviors in a cluster as outliers. In particular, a fast normalized longest common subsequence (nLCS) is used as the similarity measure for comparing sequences, instead of computational expensive LCS measure. Furthermore, the authors provide a detailed analysis for detected sequence outliers and also reveal possible anomalous events inside these sequences.

Sun et al. [78] propose an approach for detecting outliers in sequence data sets depending on building a probabilistic suffix tree (PST), which exploits the theory of a variable-order markov chain and uses a suffix tree as its index structure. In detecting outliers, only nodes near the root of the tree need to be examined for distinguishing outliers from non-outliers. This helps to reduce the size and time of the construction of PST. The approach uses the length of *normalized probability* as the sequence similarity measure, which can efficiently find the top n outliers in a sequence data set, and also determine a given query sequence is an outlier if its value of similarity measure is greater than a user-defined threshold. The theory of *entropy* further verifies the effectiveness of the similarity measure.

5.3.1 *Evaluation of Techniques for Sequence Data Sets.* Both clustering-based and tree-based approaches use efficient similarity measures for comparing sequences and further distinguish abnormal subsequences from non-outliers. Differently, clustering-base approaches use traditional clustering algorithms to cluster the sequence data into groups. Tree-based approaches construct a specific tree as index to decompose data structure. However, these approaches may be suspectable to high dimensional data sets. Also, time complexity will increase with the construction of trees.

5.4 Outlier Detection Techniques for Spatial Data Set

Spatial data has non-spatial and spatial attributes. Non-spatial attributes include the intrinsical information of data characteristic and are used to compare and distinguish spatial objects in spatial neighborhoods. In traditional outlier detection techniques, distribution-based approaches work well in one-dimensional data sets and only consider the statistical distribution of non-spatial attribute values. They

ignore the spatial relationships between data points. On the other hand, most non-parametric methods do not distinguish between spatial and non-spatial attributes, but use all dimensions to define the neighborhood based on concepts of distance, density and convex-hull depth. Thus, many spatial outlier detection approaches have been developed and can be categorized into *graphical* and *statistical* approaches. The first category uses a graph to illustrate the distribution of neighborhood difference and identify those points in particular positions of the graph as outliers. These approaches include variogram clouds [79], pocket plots [80], scatterplot [31] and Moran scatterplot [81]. Here, we focus on several spatial outlier detection techniques [82, 83, 84, 85, 86, 87, 88, 89, 90] in statistical approaches.

Shekhar et al. [82] originally introduce a general definition of a spatial outlier, i.e., *a spatially referenced point whose non-spatial attribute values are significantly different from those of other spatially referenced points in its spatial neighborhood.* This definition indicates whether a data point is declared as a spatial outlier depends on the difference between an attribute value of the point and the average attribute value of its spatial neighbors. The proposed algorithm uses a *single* non-spatial attribute to compare the difference between spatial neighborhoods and identifies spatial outliers computationally efficiently by computing the global algebraic aggregate functions. Shekhar et al. [83] further consider the graph structure of the spatial data and exploit a *graphical* method for spatial outlier detection. Specifically, this method compares the difference between an attribute value of a data point and the average attribute of its neighbors, and checks whether that the whole data fits into a normal distribution. Those points whose testing values are greater than a specified confidence interval are considered as spatial outliers.

Lu et al. [84] propose two *iterative* algorithms and a *non-iterative* algorithm to detect spatial outliers. All of three algorithms depend on the choice of a neighborhood function and a comparison function. The neighborhood function refers to a summary statistic of attribute values of all the spatial neighbors of a data point. The comparison function is used to further compare the attribute value of this point with the summary statistic value of its neighbors. In particular, two iterative algorithms use the same neighborhood function and identify only one outlier through each iteration, and then the attribute value of this outlier is immediately replaced by the average attribute value of its neighbors so that this outlier will not impact the subsequent iterations negatively. The non-iterative algorithm defines a different neighborhood function based on the *median* of the attribute values of the neighbors. The advantage of the use of median is to reduce the negative impact caused by the presence of neighboring points with extremely high/low attribute values. Lu et al. [85] further detect spatial outliers with *multiple attributes* using Mahalanobis distance, which considers ranges of variance and covariance between attributes measured. The comparison function is assumed to follow a multivariate normal distribution, and those points can be considered as spatial outliers if their Mahalanobis distance is larger than a specified confidence interval.

Kou et al. [86] present two spatial *weighted* outlier detection algorithms, which considers the impact of spatial relationship on the neighborhood comparison. For a data point, each of neighbors in its spatial neighborhood is assigned a different weight in terms of their own impact on the point. In the first algorithm, a weighted

average of the non-spatial attribute value of its neighbors is computed to compare the difference with the attribute value of this point. The top n point with the largest difference values are identified as outliers. The second algorithm compares the attribute value of the point with each of its neighbors one by one, which helps to retain the variance among the neighbors.

Chawla and Sun [87] propose a spatial *local* outlier detection approach. The issue of detecting spatial local outliers is usually missed by most techniques, which focus on discovering spatial global outliers by the statistical test. This approach, based on LOF [18], efficiently computes the degree of outlierness of each point in a data set, and also considers the values of *spatial autocorrelation* and *spatial heteroscedasticity*, which are used to capture the effect of a data point on its neighborhood and the non-uniform variance of the data, respectively. The use of the two statistics values aims to avoid the assumption of an identical or independent distribution of data.

Zhao et al. [88] present a *wavelet analysis* based spatial outlier detection approach to detect *region* outliers, which aims at determining a group of outliers instead of an outlier. The multi-resolution character of wavelet analysis enables that the original data is decomposed into different spatial scales, and then the localization of variation shown in the frequency domain further determines the spatial regions of the potential spatial outliers at certain scales of interest. Each suspect region outlier is further viewed as a single point and is verified whether it is a true outlier by the spatial outlier detection algorithm proposed in [85]. Lu et al. 2004 [89] extend this work and identify the boundary of region outliers using edge detection with competitive fuzzy classifier.

Adam et al. [90] address a definition of *micro neighborhood* based on both the spatial relationship and the semantic relationship among neighbors. Based on this definition, a point can be considered as an outlier if it differs sufficiently from other points in the micro neighborhood. The authors use the same distance-based outlier detection technique proposed in [42] to identify spatial outliers.

5.4.1 *Evaluation of Techniques for Spatial Data Sets.* Graph-based approaches illustrate the distribution of neighborhood difference and identify those points in particular positions of the graph as outliers. Statistical approaches explicitly distinguish between spatial and non-spatial attributes, and effectively identify outliers by comparing the non-spatial attribute of a point with the average non-spatial attribute value of its spatial neighbors. However, they assume a normal distribution to represent the whole data, which is not always the case in many real-life situations.

5.5 Outlier Detection Techniques for Streaming Data Set

Traditional outlier detection techniques work well in static data sets, where all data points are stationary. However, in streaming and dynamic data sets, a large volume of data is continuously and fast transferred in an ordered sequence, and also data may be constantly added, removed, or updated. In this data set, a data model built in a particular time instant may be invalid in consequent time instants. Thus, data stream can be viewed as an infinite process consisting of data which continuously evolves with the time [91]. The issue of detecting outliers in streaming data has gained much attention and several relevant approaches are categorized into *model-based* [91, 72, 92], *graph-based* [93, 94] and *density-based* [95, 96, 97] methods.

He et al. [91] present a model-based approach to identify outliers in data streams by using *frequent patterns*, which represent common patterns of a majority of data points in data sets. The degree of outlierness for each point is measured by a frequent pattern outlier factor (FPOF) and n points that contain very few frequent patterns are considered as outliers. In particular, a *lossy counting* technique is used to efficiently approximate frequent patterns over data streams with a single pass, instead of finding them exactly with multiple passes. However, the authors do not give any experimental results or performance evaluation of this approach.

The outlier detection approach proposed by Otey et al. [72] also can handle dynamic and streaming data in a distributed setting. Specifically, a local model is first built in each participating local data set, and a global model is constructed based on these local models. As a result, each local data set has a copy of the global model and can calculate the degree of outlierness for each point in its local data set by an anomaly score function. Points whose score are greater than a pre-defined threshold are flagged as local outliers.

Yamanishi et al. [92] detect outliers in non-stationary time series data based on a typical statistical autoregression (AR) model, which represents a statistical behavior of time series data. The proposed approach first incrementally learns the AR model by a sequentially discounting AR model estimation (SDAR) algorithm, which can effectively deal with non-stationary data streams by gradually weakening the effect of past data. Each data point is then assigned to an anomaly score, which measures how large the point deviates from the learned AR model. A higher score of a point indicates that the point is more likely to be an outlier.

Muthukrishnan et al. [93, 94] define a new notion of an outlier in time series data streams based on a representation sparsity metric *histogram*, i.e., *if the removal of a point from the time sequence results in a sequence that can be represented more briefly than the original one, then the point is an outlier*. In the histogram, a few *buckets* partitioned from data stream can summarize the data. Outlier are points whose removal from the data set result in an improved succinct representation of the remaining data. The proposed approach can efficiently find outliers at any instant of massive data streams over time only using very small memory space.

Pokrajac et al. [95] propose an incremental density-based approach to detect outliers in data streams. This approach exploits the static *iterated* LOF [18] algorithm to deal with each new point inserted into the data set and iteratively determines whether the point is an outlier. Furthermore, it can efficiently adapt to the update of the data profiles caused by insertion or deletion of data points. Experimental results show that the incremental LOF has the equivalent detection performance with LOF and is also more computationally efficient. However, it is not appropriate for high dimensional data because of expensive computation of indexing structures.

Palpanas et al. [96] propose an original outlier detection approach for real-time streaming sensor data in wireless sensor networks (WSNs). This approach has no a priori knowledge of data distribution, but instead builds a model of the most recent values in a *sliding window* and also uses *kernel density estimator* to efficiently and effectively approximate the data distribution. Outliers are those values that deviate significantly from the model by using the distance-based approach in [42]. Subramaniam et al. [97] further use LOCI [51] to identify *density-based* outliers,

which overcomes the disadvantages of distributed-based outlier detection and also being applied to the multi-dimensional data. Experimental results show that this approach achieves high accuracy of data distribution estimation and high outlier detection rate with low memory occupation and message transmission.

5.5.1 *Evaluation of Techniques for Streaming Data Sets.* Model-based approaches represent the statistical behavior of data stream by the construction of a model and further declare those points that deviate from this model as outliers. They can efficiently deal with the streaming data in an online fashion, however, it is not easy to construct an accurate model to represent the whole data. Graph-based approaches identify outliers based on estimating the distribution of data stream. However, they may suffer from the curse of dimensionality and the accuracy of the estimation of data distribution. Density-based approaches overcome the disadvantages of distribution-based outlier detection and also are applied to the multi-dimensional data. They can achieve high outlier detection rate with low memory occupation.

5.6 Outlier Detection Techniques for Spatio-Temporal Data Set

Most existing spatial outlier detection techniques focus on detecting spatial outliers, which only considers the non-spatial attributes of data or the spatial relationships among neighbors. However, in all geographic phenomena evolving over time, temporal aspects and spatial-temporal relationships existing among spatial data points also need to be considered in detecting outliers. Currently, two initial work [98, 99] have addressed the detection of spatio-temporal outliers in data sets on the basis of *clustering* concepts and *statistical* tests.

Cheng and Li [98] introduce a formal definition of spatio-temporal outliers (ST-outliers), i.e., *a spatial-temporal point whose non-spatial attribute values are significantly different from those of other spatially and temporally referenced points in its spatial or/and temporal neighborhoods.* The definition indicates that ST-outliers are identified by comparing the spatio-temporal points with their spatio-temporal neighbors. Considering the temporal aspects, the authors declare a point as a ST-outlier by checking if the point's attribute value at time T is significantly different from the statistical attribute values of its neighbors at time $T-1$ and $T+1$. They further propose a four-step approach to detect ST-outliers, i.e., classification, aggregation, comparison and verification. In particular, the classification step aims at finding out the spatio-temporal points of interest by *clustering* the input data, which can be achieved by either supervised classification based on priori knowledge of the data or unsupervised classification if prior knowledge of data is not available. The aggregation step uses different spatial scales of the data to generate different clusters and effectively filter the noises. In comparison step, potential spatial outliers can be identified by comparing the results obtained from the classification step with the results obtained from the aggregation step. The verification step further compares these potential spatial outliers with their temporal neighbors in a continuous pattern. If the difference value is greater than a *statistical* threshold, these outliers will be considered as true ST-outliers. Experimental results generated from a practical coastal geomorphic study show the effectiveness of the approach. However, the author do not consider the effect of the granularity of spatial and temporal scales, and also the temporal dimension is poorly treated without any metrics.

Birant and Kut [99] define a similar definition of ST-outlier as [98] and present a ST-outlier detection approach based on clustering concepts. In particular, this approach consists of three steps, clustering, checking spatial neighbors, and checking temporal neighbors. In the clustering step, an efficient *clustering* technique DBSCAN [33] has been improved in supporting temporal aspects and detecting outliers in clusters with different densities. As a result, potential outliers are those points which do not belong to any of clusters. The following two steps further verify these potential outliers. In the checking spatial neighbors step, a potential outlier is labelled as a spatial outlier if its *statistic* value is outside a user-specified confidence interval. In the checking temporal neighbors step, if this spatial outlier is significantly different from its temporal neighbors in consecutive time units, it is labelled as a true ST-outlier. This approach uses several pre-defined parameters and some of them are very sensitive for the performance of outlier detection. Also, Euclidean distance is used to compute the distance of any two points, which is not suitable for high dimensional data sets.

5.6.1 *Evaluation of Techniques for Spatio-Temporal Data Sets.* Clustering-based approaches identify the spatio-temporal points of interest in clusters with different densities and further verify these potential outliers by statistical tests. They can explicitly consider the temporal aspects and spatial-temporal relationships existing among spatial data points during outlier detection. However, the temporal dimension is poorly treated without any appropriate metrics so that these approaches are susceptible to high dimensional data sets. Also, they assume a normal distribution to represent the whole data, which is not available in many real-life situations.

6. DECISION TREES

Availability of various outlier detection techniques with different characteristics makes selecting the most suitable technique for a specific data sets more crucial than ever. In this section, we present two decision trees to be used as guidelines. Table II gives an overview of appropriate techniques for application domains. Table III & IV compare characteristics of outlier detection techniques based on type of data sets.

Table II. Outlier detection techniques applied to specific application domains

Technique-based on	Application domains					
	Fraud detection	Intrusion detection	Environmental monitoring	Medical and public health	Localization and tracking	Logistics and transportation
Distribution	√	√	√	√	√	√
Depth	√	√	√	√	√	√
Graph	√	√	√	√	√	√
Clustering	√	√	√	√	√	
Distance		√	√	√	√	
Density		√	√	√	√	
NN	√	√		√		
SVM		√				
Subspace		√				
Tree	√	√		√		
Model		√	√	√	√	√

Table III. Classification and comparison of outlier detection techniques for simple data sets

Name of the technique/ authors	Outliers property				Outlier detection technique property			Data sets property			
	Outlier type		Outlier degree		Technique-based on	Number of outliers detected at once		Data dimension			Data type
	Global	Local	Scalar	Outlier-ness		One	Multiple	Univariate	Multivariate		
									Moderate	High	
Grubbs *et al.* [18], 1969	√		√		Distribution	√		√			Numeric
Barnett *et al.* [5], 1994	√		√		Distribution		√	√			Numeric
Eskin *et al.* [23], 2000	√		√		Distribution	√		√			Numeric
Yamani. *et al.* [24], 2000	√			√	Distribution		√		√		Mixed-type
MVE [19], 1996	√		√		Depth		√		√		Numeric
Convex peeling [19], 1996	√			√	Depth	√			√		Numeric
Box plot [29], 2000	√		√		Graph		√	√			Mixed-type
Scatter plot [30], 1996	√		√		Graph		√		√		Mixed-type
Yu *et al.* [38], 2002	√		√		Clustering		√		√		Numeric
Jiang *et al.* [39], 2001	√			√	Clustering	√			√		Numeric
He *et al.* [40], 2003		√		√	Clustering		√		√		Numeric
Ren *et al.* [41], 2004	√			√	Clustering		√		√		Numeric
Knorr *et al.* [42], 1998	√		√		Distance		√		√		Numeric
Ramas. *et al.* [44], 2000	√			√	Distance		√		√		Numeric
Bay *et al.* [45], 2003	√			√	Distance		√		√		Numeric
LOF [46], 2000		√		√	Density		√		√		Numeric
CF [47], 2003		√		√	Density		√		√		Numeric
Jin *et al.* [48], 2001		√		√	Density		√		√		Numeric
Tang *et al.* [49], 2002		√		√	Density		√		√		Numeric
Hu *et al.* [50], 2003		√		√	Density		√		√		Numeric
LOCI [51], 2003		√		√	Density		√		√		Numeric
Kim *et al.* [52], 2006		√		√	Density		√		√		Numeric
RDF [53], 2004		√	√		Density		√		√		Numeric
Fan *et al.* [54], 2006		√		√	Density		√		√		Numeric
Kollios *et al.* [55], 2003		√		√	Density		√		√		Numeric
Harkins *et al.* [57], 2002	√			√	NN		√		√		Numeric
Fu *et al.* [58], 2006	√		√		NN		√		√		Numeric
Scholko. *et al.* [61], 2001	√		√		SVM		√		√		Numeric
Tax *et al.* [62], 1999	√		√		SVM		√		√		Numeric
Petrovskiy [63], 2003	√			√	SVM		√		√		Numeric

Table IV. Classification and comparison of outlier detection techniques for complex data sets

Name of the technique/ authors	Outliers property				Outlier detection technique property			Data sets property			
	Outlier type		Outlier degree		Technique-based on	Number of outliers detected at once		Data dimension			Data type
	Global	Local	Scalar	Outlier-ness		One	Multiple	Univariate	Multivariate		
									Moderate	High	
Aggarwal *et al.* [64], 2005	√			√	Subspace		√			√	Numeric
Zhu *et al.* [66], 2005	√			√	Subspace		√			√	Numeric
Shyu *et al.* [67], 2003	√			√	Subspace		√			√	Numeric
Angiulli *et al.* [69], 2005	√			√	Distance		√			√	Numeric
Ghoting *et al.* [70], 2006	√			√	Distance		√			√	Numeric
Chaudha. *et al.* [71], 2002	√			√	Distance		√		√		Numeric
Otey *et al.* [72], 2006	√			√	Graph		√		√		Mixed-type
Wei *et al.* [74], 2003		√		√	Graph		√		√		Mixed-type
He *et al.* [75], 2005		√		√	Graph	√			√		Mixed-type
Yu *et al.* [76], 2006		√		√	Graph		√		√		Mixed-type
Budalak. *et al.* [77], 2006	√			√	Clustering		√		√		Sequence
Sun *et al.* [78], 2006	√			√	Tree		√		√		Sequence
Shekhar *et al.* [82], 2001	√		√		Distribution		√	√			Spatial
Lu *et al.* [84], 2003	√		√		Distribution	√		√			Spatial
Lu *et al.* [85], 2003	√		√		Distribution		√		√		Spatial
Kou *et al.* [86], 2006	√			√	Distribution		√		√		Spatial
Sun *et al.* [87], 2004		√		√	Distribution		√		√		Spatial
Zhao *et al.* [88], 2003		√	√		Distribution		√		√		Spatial
He *et al.* [91], 2003	√			√	Model		√		√		Streams
Yamani. *et al.* [92], 2006	√		√		Model		√		√		Streams
Muthukri. *et al.* [93], 2004	√		√		Graph		√		√		Streams
Pokrajac *et al.* [95], 2007		√		√	Density		√		√		Streams
Subrama. *et al.* [97], 2006		√		√	Density		√		√		Streams
Cheng *et al.* [98], 2006	√		√		Clustering & Distribution		√		√		Spatial-temporal
Birant *et al.* [99], 2006	√		√		Clustering & Distribution		√		√		Spatial-temporal

7. CONCLUSION

In this paper, we present a comprehensive taxonomy framework for contemporary outlier detection techniques based on simple and specific semantics of data. Also, we introduce a through performance evaluation of each class of outlier detection techniques under this taxonomy framework. Furthermore, we provide two decision trees to choose suitable outlier detection techniques for specific application domains and type of data sets.

There is no single universally applicable or generic outlier detection approach. Thus, the developers should consider whether an outlier detection technique is suitable for a data set depending on several important aspects, i.e., the use of pre-labelled data, the use of parameters of data distribution, the type and dimension of detected outliers, the degree of being outliers, the number of detected outliers at once. Also, outlier detection techniques are desired to maintain a high detection rate while keeping the false alarm rate and time complexity low. Additionally, the number of used user-defined parameters should be taken into account.

REFERENCES

[1] Pang-Ninh Tan (2006) Knowledge Discovery from Sensor Data. Sensors

[2] Pei Sun (2006) Outlier detection in high dimensional, spatial and sequential data sets. Doctoral Thesis

[3] V. J. Hodge, J. Austin (2003) A survey of outlier detection methodologies. Artificial Intelligence Review, vol. 22, pp 85-126

[4] D.M. Hawkins (1980) Identification of outliers. Chapman and Hall, Reading, London

[5] V. Barnett and T. Lewis (1994) Outliers in statistical data. John Wiley Sons, Reading, New York

[6] J. Han and M. Kamber (2001) Data mining: concepts and techniques. Morgan Kaufmann

[7] R. J. Bolton, D. J. Hand (2001) Unsupervised profiling methods for fraud detection. In: Proceedings of CSCC

[8] D.J. Marchette (2001) Computer intrusion detection and network monitoring: a statistical viewpoint. New York: Springer

[9] G. M. Davis, K. B. Ensor (2006) Outlier detection in environmental monitoring network data: an application to ambient ozone measurements for Houston, Texas. Journal of Statistical Computation and Simulation, vol. 76, no. 5, pp 407-422

[10] J. Lin, A. E. Fu and H. V. Herle (2005) Approximations to magic: Finding unusual medical time series. In: Proceedings of Symposium on Computer-Based Medical systems. Washington, DC, USA, pp 329-334

[11] W. Du, L. Fang and P. Ning (2005) LAD: localization anomaly detection for wireless sensor networks. In: Proceedings of Parallel and Distributed Processing Symposium

[12] R. Milen, A. Sohal and S. Moss (1999) Quality management in the logistics function: an empirical study. Journal of Quality & Reliability Mangement, Volume 16, pp 166-180

[13] M. Markos, S. Singh (2003) Novelty detection: a review-part 1: statistical approaches. Signal Processing, vol. 83, pp 2481-2497

[14] M. Markos, S. Singh (2003) Novelty detection: a review-Part 2: neural network based approaches. Signal Processing, vol. 83, pp 2499-2521

[15] P-N Tan, M.Steinback, V.Kumar (2005) Introduction to data mining. Addison Wesley

[16] R. E. Bellman (1961) Adaptive control processes: a guided tour. Princeton University Press

[17] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar (2003) A comparative study of anomaly detection schemes in network intrusion detection. In: Proceedings of SIAM

[18] Grubbs, Frank (1969) Procedures for detecting outlying observations in samples. Technometrics, vol. 11, no. 1, pp. 1-21

[19] P.J. Rousseeuw and A .M. Leroy (1996) Robust regression and outlier detection. John

Wiley and Sons
[20] N. Ye and Q. Chen (2001) An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. Quality and Reliability Engineering International, vol. 17, no. 2, pp.105-112
[21] J. Hardin, D.M. Rocke (2004) Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. Computational Statistics and Data Analysis, vol 44, pp. 625-638
[22] P. J. Rousseeuw, K.V. Driessen (1999) A fast algorithm for the minimum covariance deteriminant estimator. Technometrics, vol. 41, pp 212-223
[23] E. Eskin (2000) Anomaly detection over noisy data using learned probability distributions. In: Proceedings of Machine Learning
[24] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne (2000) On-line unsupervised learning outlier detection using finite mixtures with discounting learning algorithms. In: Proceedings of KDD, pp 250-254
[25] F. Preparata and M. Shamos (1988) Computational geometry: an introduction. Springer, Berlin Heidelberg, Reading, New York
[26] J. Tukey (1997) Exploratory data analysis. Addison-Wesley
[27] I. Ruts and P. Rousseeuw (1996) Computing depth contours of bivariate point clouds. Journal of Computational Statistics and data Analysis, 23:153-168
[28] T. Johnson, I. Kwok, and R.T. Ng. (1998) Fast computation of 2-dimensional depth contours. In: Proceedings of KDD, pages 224-228
[29] J. Laurikkala, M. Juhola, E. Kentala (2000) Informal identification of outliers in medical data. In: Proceedings of IDAMAP
[30] Y. Panatier (1996) Variowin. Software for spatial data analysis in 2D. Springer-Verlag, New York
[31] P. M. Valero-Mora, F. W. Young, M. Friendly (2003) Visualizing categorical data in ViSta. Computational Statistics & Data Analysis, vol. 43, pp 495-508
[32] D. J. Marchette, J. L. Solka (2003) Using data images for outlier detection. Computational Statistics & Data Analysis, vol. 43, no. 4, pp 541-552
[33] Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of KDD, pp 226-231
[34] Karypis G, Han E, Kumar V (1999) Chameleon: a hierarchical clustering algorithm using dynamic modeling. IEEE Computing 32(8):68-75
[35] Zhang T, Ramakrishnan R, LivnyM (1996) BIRCH: an efficient data clustering method for very large databases. In: Proceedings of ACM SIGMOD, pp 103-114
[36] Guha S, Rastogi R, Shim K (1998) CURE: an efficient clustering algorithm for large databases. In: Proceedings of ACM SIGMOD, pp 73-84
[37] A. Foss and O. Zaane (2002) A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets. In: Proceedings of ICDM
[38] D. Yu, G. Sheikholeslami, and A. Zhang (2002) Findout: finding outliers in very large datasets. In: Journal of Knowledge and Information Systems, vol. 4, no. 3, pp. 387-412
[39] M. F. Jiang, S. S. Tseng, C. M. Su (2001) Tw-phase clustering process for outliers detection. Pattern Recognition Letters, 22 (6-7): 691-700
[40] Z. He, X. Xu, S. Deng (2003) Discovering cluster based local outliers. Pattern Recognition Letters, 24 (9-10): 1651-1660
[41] D. Ren, I. Rahal, W. Perrizo (2004) A vertical outlier detection algorithm with clusters as by-product. In: Proceedings of ICTAI
[42] Knorr E, Ng R(1998) Algorithms for mining distance-based outliers in large data sets. In: Proceedings of VLDB, pp 392-403
[43] E. Knorr, R. Ng, and V. Tucakov (2000) Distance-based outlier: algorithms and applications. Journal of VLDB, vol. 8, no. 3-4, pp 237- 253
[44] Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: Proceedings of ACM SIGMOD, pp 427-438
[45] S. Bay and M. Schwabacher (2003) Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proceedings of ACM SIGKDD, pp 29-38
[46] Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) LOF: identifying density-based local

outliers. In: Proceedings of ACM SIGMOD, pp 93-104
[47] A. L. Chiu, A. W. Fu (2003) Enhancements on local outlier detection. In: Proceedings of IDEAS
[48] W. Jin, A.K.H. Tung, and J. Han (2001) Mining top-n local outliers in large databases. In: Proceedings of ACM SIGKDD, pp. 293-298
[49] J. Tang, Z. Chen, A. W. Fu, F. W. Cheung (2002) Enhancing effectiveness of outlier detections for low density patterns. In: Proceedings of PAKDD, pp 535-548
[50] T. Hu, S. Y. Sung (2003) Detecting pattern-based outliers. Pattern Recognition Letters, 24 (16): 3059-3068
[51] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. (2003) LOCI: fast outlier detection using the local correlation integral. In: Proceedings of ICDE, pp 315-326
[52] S. Kim, S. Cho (2006) Prototype based outlier detection. In: Proceedings of IJCNN, pp 820-826
[53] D. Ren, B. Wang, W. Perrizo (2004) RDF: a density-based outlier detection method using vertical data representation. In: Proceedings of ICDM, pp 503-506
[54] H. Fan, O. R. Zaiane, A. Foss, J. Wu (2006) A nonparametric outlier detection for effectively discovering top-n outliers from engineering data. In: Proceedings of PAKDD, pp 557 - 566
[55] G. Kollios, D. Gunopulos, N. Koudas, S. Berchtold (2003) Efficient biased sampling for approximate clustering and outlier detection in large data sets. Knowledge and Data Engineering, vol. 15, no. 5, pp 1170-1187
[56] P. Sykacek (1997) Equivalent error bars for neural network classifiers trained by bayesian inference. In: Proceedings of ESANN
[57] S. Harkins, H. He, G. J. Willams, R. A. Baster (2002) Outlier detection using replicator neural networks. In: Proceedings of DaWaK, pp 170-180
[58] Fu. J, X. Yu (2006) Rotorcraft acoustic noise estimation and outlier detection. In: Proceedings of IJCNN, pp 4401-4405
[59] G. J. Willams, R. A. Baster, H. He, S. Harkins, L. Gu (2002) A comparative study of RNN for outlier detection in data mining. In: Proceedings of ICDM, pp 709-712
[61] B. Schlkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson (2001) Estimating the support of a high dimensional distribution. Neural Computation, vol. 13, no. 7, pp 1443-1471
[62] D. M. J. Tax and R. P. W. Duin (1999) Support vector domain description. Pattern Recognition Letters, vol. 20, pp 1191-1199
[63] M. I. Petroveskiy (2003) Outlier detection algorithms in data mining system. Programming and Computer Software, vol. 29, no. 4, pp 228-237
[64] C. C. Aggarwal, S. P. Yu (2005) An effective and efficient algorithm for high dimensional outlier detection. VLDB Journal, vol. 14, pp 211-221
[65] Charu C. Aggarwal and Philip S. Yu (2001) Outlier detection for high dimensional data. In: Proceedings of ACM SIGMOD, pp 37-47
[66] C. Zhu, H. Kitagawa, C. Faloutsos (2005) Example-based robust outlier detection in high dimensional datasets. In: Proceedings of ICDM, pp 829-832
[67] M. Shyu, S. Chen, K. Sarinnapakorn, L. W. Chang (2003) A novel anomaly detection scheme based on principal component classifier. In: Proceedings of ICDM, pp172-179
[68] F. Angiulli and C. Pizzuti (2002) Fast outlier detection in high dimensional spaces. In: Proceedings of PKDD, pp 15-26
[69] F. Angiulli, C. Pizzuti (2005) Outlier mining in large high-dimensional data sets. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 2, pp 203-215
[70] A. Ghoting, S. Parthasarathy, M. Otey (2006) Fast mining of distance-based outliers in high dimensional datasets. In: Proceedings of SIAM
[71] A. Chaudhary, A. S. Szalay, and A. W. Moore (2002) Very fast outlier detection in large multidimensional data sets. In: Proceedings of ACM SIGMOD Workshop on DMKD
[72] M. E. Otey, A. Ghoting, S. Parthasarathy (2006) Fast distributed outlier detection in mixed-attribute data sets. Data Mining and Knowledge Discovery, vol. 12, no. 2-3, pp 203-228
[73] A. Ghoting, M. Otey, S. Parthasarathy (2004) LOADED: link-based outlier and anomaly detection in evolving data sets. In: Proceedings of Proceedings of ICDM, pp 387-390
[74] L. Wei, W. Qian, A. Zhou, W. Jin, J. X. Yu (2003) HOT: hypergraph-based outlier test

for categorical data. In: Proceedings of PAKDD, pp 399-410
[75] Z. He, S. Deng, X. Xu (2005) An optimization model for outlier detection in categorical data. In: Proceedings of ICIC, pp 400-409
[76] J. X. Yu, W. Qian, H. Lu, A. Zhou (2006) Finding centric local outliers in categorical/numerical spaces. Knowledge Information System, vol. 9, no. 3, pp 309-338
[77] S. Budalakoti, S. Cruz, A. N. Srivastava, R. Akella, E. Turkov (2006) Anomaly detection in large sets of high-dimensional symbol sequences. NASA TM
[78] P. Sun, S. Chawla, B. Arunasalam (2006) Mining for outliers in sequential databases. In: Proceedings of SIAM, pp 94-105
[79] J. Haslett, R. Brandley, P. Craig, A. Unwin, G. Wills (1991) Dynamic graphics for exploring spatial data with application to locating global and local anomalies. The American Statistician, 45:234-242
[80] A. Luc (1995) Local indicators of spatial association: LISA. Geographical Analysis, 27(2): 93-115
[81] R. Haining (1993) Spatial data analysis in the social and environmental sciences. Cambridge University Press
[82] S. Shekhar, C.-T. Lu, and P. Zhang (2001) A unified approach to spatial outliers detection. GeoInformatica, 7(2): 139-166
[83] S. Shekhar, C.-T. Lu, and P. Zhang (2001) Detecting graph-based spatial outlier: algorithms and applications(a summary of results). In: Proceedings of ACM SIGKDD, pp 371-376
[84] C.-T. Lu, D. Chen, and Y. Kou (2003) Algorithms for spatial outlier detection. In: Proceedings of ICDM, pp 597-600
[85] C.T. Lu, D. Chen, and Y. Kou (2003) Detecting spatial outliers with multiple attributes. In: Proceedings of ICTAI, pp 122-128
[86] Y. Kou, C. Lu, D. Chen (2006) Spatial weighted outlier detection. In: Proceeding of SDM
[87] P. Sun, S. Chawla (2004) On local spatial outliers. In: Proceedings of ICDM, pp 209-216
[88] J. Zhao, C.-T. Lu, and Y. Kou (2003) Detecting region outliers in meteorological data. In: proceedings of ACM GIS, pp 49-55
[89] C. Lu and L.R. Liang (2004) Wavelet fuzzy classification for detecting and Tracking region outliers in meteorological data. In: Proceedings of GIS, pp 258-265
[90] N. R. Adam, V.P. Janeja, V. Atluri (2004) Neighborhood based detection of anomalies in high dimensional spatiotemporal sensor datasets. In: Proceedings of ACM Symposium on Applied Computing, pp 576-583
[91] Z. He, X. Xu, S. Deng (2003) Outlier detection over data streams. In: Proceedings of ICYCS
[92] K. Yamanishi, J. Takeuchi (2006) A unifying framework for detecting outliers and change points from non-stationary time series data. Knowledge and Data Enginnering, vol. 18, no. 4, pp 482-492
[93] S. Muthukrishnan, R. Shah, J. S. Vitter (2004) Mining deviants in time series data streams. In: Proceedings of SSDBM
[94] H. V. Jagadish, N. Koudas, S. Muthukrishnan (1999) Mining deviants in a time series database. VLDB
[95] D. Pokrajac, A. Lazarevic, L. J. Latechi (2007) Incremental local outlier detection for data streams. In: Proceedings of CIDM
[96] T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos (2003) Distributed deviation detection in sensor networks. In: Proceedings of ACM SIGMOD, pp 77-82
[97] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos (2006) Online outlier detection in sensor data using non-parametric models. VLDB
[98] T. Cheng and Z. Li (2006) A multiscale approach for spatio-temporal outlier detection. Transactions in GIS, vol. 10, no. 2, pp 253-263
[99] D. Birant, A. Kut (2006) Spatio-temporal outlier detection in large database", In: Proceedings of ITI
[100] V. Chandola, A. Banerjee and V. Kumar (2007) Outlier detection: a survey, Technical Report. Univeristy of Minnesota, USA