Research Journal of Applied Sciences, Engineering and Technology 8(3): 398-409, 2014
DOI:10.19026/rjaset.8.986
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: March 29, 2014 Accepted: April 28, 2014 Published: July 15, 2014

Research Article

Rough K-means Outlier Factor Based on Entropy Computation

Djoko Budiyanto Setyohadi, Azuraliza Abu Bakar and Zulaiha Ali Othman
Data Mining and Optimization Research Group, Center for Artificial Intelligence Technologi,
Faculty of Information Science and Technologi, Universiti Kebangsaan Malaysia, Bangi,
Selangor DarulEhsan, 43000, Malaysia

Abstract: Many studies of outlier detection have been developed based on the cluster-based outlier detection approach, since it does not need any prior knowledge of the dataset. However, the previous studies only regard the outlier factor computation with respect to a single point or a small cluster, which reflects its deviates from a common cluster. Furthermore, all objects within outlier cluster are assumed to be similar. The outlier objects intuitively can be grouped into the outlier clusters and the outlier factors of each object within the outlier cluster should be different gradually. It is not natural if the outlierness of each object within outlier cluster is similar. This study proposes the new outlier detection method based on the hybrid of the Rough K-Means clustering algorithm and the entropy computation. We introduce the outlier degree measure namely the entropy outlier factor for the cluster based outlier detection. The proposed algorithm sequentially finds the outlier cluster and calculates the outlier factor degree of the objects within outlier cluster. Each object within outlier cluster is evaluated using entropy cluster-based to a whole cluster. The performance of the algorithm has been tested on four UCI benchmark data sets and show outperform especially in detection rate.

Keywords: Entropy outlier, outlier detection, rough k-means

INTRODUCTION

An outlier data is the data, which is considerably different from the rest of the data in the entire data set (Hawkins, 1980). In fact, outlier data can be generated intentionally or unintentionally. An abnormal condition in medical data, defect condition of the product is an example how the outliers are generated. In many data mining applications, detecting the rare instances or the outliers can be more interesting than finding the common patterns. Outlier detection in data mining has applications, including credit card fraud detection, discovery of criminal activities in electronic commerce, weather prediction, marketing and customer segmentation.

Different approaches and algorithms have been introduced to solve the outlier detection problem. They vary between statistical, distance-based, density based approaches, supervised and unsupervised learning techniques, neural networks and machine learning techniques (Hodge and Austin, 2004). The variety development of outlier detection are purposed to handling several data mining issues such as scalability, dynamic data streams, accuracy of detection and uncertainty in data that effects the performance of detection. Literature on these research can be generally classified into four major categories based on the used techniques, i.e., Distribution based approach, distance-based approach, density-based approach and clustering-based approach.

Distribution based approach is the method which explore the statistical computation. In this approach, the r the distribution assumed to fit the dataset and then the objects are evaluated whether those objects are outliers or not based on its fit the underlying model of the data. This approach is good but impractical since it needs prior data distribution and the high computation cost. Distance-based approach is developed based the perspective of distance related metrics and mostly distance based outlier detection approaches are detected based upon the concepts of local neighbourhood. Thus, the distance between data points is needed to be computed. Consequently, when the data dimensionality increases, two problems comes i.e., it becomes increasingly difficult to specify an appropriate circular local neighbourhood and it faces to the curse of dimensionality problem. Density-based approach is performed by calculating the local density of the point being investigated and the local densities of its nearest neighbours. Therefore, Density-based approach generally more effective than the distance-based but it suffer high execution time The last approaches, Cluster

Corresponding Author: Djoko Budiyanto Setyohadi, Data Mining and Optimization Research Group, Center for Artificial Intelligence Technologi, Faculty of Information Science and Technologi, Universiti Kebangsaan Malaysia, Bangi, Selangor DarulEhsan, 43000, Malaysia

Based Outlier detection, detects outliers as points that do not lie in or located far apart from any clusters and outliers is a noise of clusters implicitly.

Originally, clustering is not dedicated to detect outlier since the outlier result is only denoted duality notion that cluster is of being an outlier and not being an outlier. However cluster based approach is promising after Jiang *et al*. (2001) proposed Outlier Finding Process (OFP) as a tool for outlier detection. OFP is performed by interpreting the cluster outlier using the cluster structure. This approach is interesting for cluster based outlier detection since it leads the possibility of Outlier Factor development based on the clusters obtained. Many cluster based outlier detection are developed in cluster based outlier detection. Since the base of outlier factor computation rely on the cluster structure, the quality of cluster significantly influences into the Cluster Based Outlier Detection (CBOD) algorithm. Furthermore, the development of the cluster based outlier detection should consider the appropriate clustering algorithm as well as outlier factor development.

Outlier Factor (OF) is conceptually started by using the distance to the kth-nearest neighbour, distance-based outliers is extended and by which outliers can be ranked (Ramaswamy *et al*., 2000). The model of computation can be adopted into CBOD easily and can extend CBOD to interpret the abnormality of the outliers cluster. The implementation of computation in CBOD is conducted as a hybrid method and performed after the clustering algorithm is finished (Jiang *et al*., 2001). This approach is able to improve the performance of outlier detection especially to cop the high dimensional data space problem by localizing the outlier factor computation. Therefore, many algorithms are developed based on this approach (He *et al*., 2004; He *et al*., 2006; Duan *et al*., 2009). He *et al*. (2004) introduced the infrequent item set of association rules mining to detect the outliers. The basic concept is to assume that the infrequent item which item is not frequent in the transaction has potential to be an outlier. Assumed, outlier is closely to the degree of disorder of a dataset; entropy is proposed to optimize the performance on the categorical data set (He *et al*., 2005). The previous methods are successful to detect outlier however they aimed to detect local cluster outlier. In fact, there are several of outlier due to this problems Duan *et al*. (2009), develop a new outlier detection which is purposed to detect both outlier point and outlier cluster sequentially.

CBOD is simple and not need any preliminary knowledge. Simple comes the fact that outlier is resulted from the product of clustering algorithm or its extension. Outlier factor generated CBOD is linearly depending on the cluster quality which reflects class or pattern within dataset. Thus the worse pattern influences significantly to the performance of outlier detection algorithm. Moreover, the use of supervised classification which is addressed to improve the performance outlier detection is an alternative to get a better pattern. The use of the supervised classification is aimed to improve the performance of detection especially when algorithm should map the pattern/structure of dataset. However, the use of supervised approach still remains two disadvantages i.e., the higher of computation cost RSetAlg (Shaari *et al*., 2009) and not all of the data is labelled. These situations motivate us to develop the CBOD algorithm by developing suitable clustering algorithm and appropriate outlier factor measure.

Our proposed approach differs from the previous algorithm. Firstly, we will treat an outlier as a point and as a group. Secondly, we extend an outlier as an object which is defined as given by Hawkins (1980) i.e., Outlier is an objects or group which deviates so much from other observations. To perform Hawkins definition, an outlier in our CBOD approach is threat as an object or group which deviate as a main measurement. Thus we will not use size as the main foundation of our outlier detection algorithm. The characteristics of outlier will be addressed by the deviation rank as a foundation of Outlier Factor measure. Consequently, our approach carries out by producing clusters which have size approximately equal compared to outlier cluster. According to the assumption that the amount of outlier objects is about 5%, logically many vague objects are produced in the overlap area of the cluster, because the bigger cluster is forced to be separated into a small cluster. As a result the overlap clustering is required to perform our algorithm.

The overlap clustering algorithm is addressed to map huge data into smaller then followed by entropy computation to measure similarity among cluster. In addition, in our algorithm, the deviation which is used to detect outlier is relies on similarity cluster or object. This approach is aimed to avoid unbalancing clustering algorithm problem and to reduce the cost of entropy computation as a basis of outlier factor calculation. In addition, we use entropy to measure the dissimilar among the clusters. So the contributions of this study are as follows: We propose a novel definition for cluster based outlier detection, which has great new intuitive appeal and numerous applications. A measure for identifying the degree of each object being an outlier is presented, which is called Entropy Based Outlier Factor (EBOF).

LITERATURE REVIEW

Clustering algorithm is aimed to group data based on their similarity. Almost all of the earlier CBOD detect outliers as the byproduct of a clustering algorithm. The basic approach of clustering algorithm to distinguish object as an outlier is by treating a smaller size cluster as an outlier (Pires and Santos-Pereira, 2005). As a consequence, in CBOD the cluster is only denoted whether cluster is outlier or not. However outliers are far more than a binary property, so a cluster-based outlier also needs an extension method to calculate its degree of being an outlier. Regarding to this extension, there are two main steps in CBOD. Firstly, a clustering algorithm is employed to cluster the data and generate the class. And then followed by Outlier Factor computation is performed.

Outlier Factor is a common measure which is used to represent the deviation degree of the object based on the characteristics dataset. There are some preliminary ideas about outlier factor. Furthermore, the outlier detection is established by measuring dissimilarity based on pattern, distance, size, or its combination. Ramaswamy *et al*. (2000) use the distance measure of the k-nearest neighbour to calculate the outlier factor. The use of distance as outlier factor is also developed by Mahoney and Chan (2003). However, this approaches susceptible to the curse of high dimensionality. Furthermore, the more effective technique is proposed by Breunig *et al*. (2000) which use density neighbourhood of local cluster to finds outliers and it well known as a local outlier since its deviation degree is measured from the local cluster. Local outlier is more meaningful rather than a binary value as defined before. However, this approach manages to detect only single point outliers. Combined both of size and distance concept of outlier on the cluster, He *et al*. (2003) proposed Cluster-Based Local Outlier Factor (CBLOF) in order to detect outlier, in order to keep out the quality of cluster. In CBLOF, after clustering is performed and then the outlier factor is calculated using the characteristics of small cluster. The implementation of distance Concept also performed in Unsupervised Intrusion Detection (CBUID) (Jiang *et al*., 2006), the outlier factor of CBUID is measured from the degree of deviation cluster the whole and is based on cluster radius threshold computation. However, these algorithms have a problem regarding the performance of clustering. The algorithms fall to find small cluster as the requirement of outlier detection. To solve this problem, Duan *et al*. (2009) analyzes the size measurement as a notion of outlier and new definition which covers both point and cluster is proposed. The new outlier detection is performed by integrating spatial and temporal locality concept. This problem is able to solve the previous CBOD problem. However the use LDBSCAN increases the time complexity of cluster based outlier detection algorithm.

In fact, the performance of clustering determines the performance of CBOD. Therefore, almost all the previous scholar develops their own clustering algorithm which is suitable into their CBOD. For example, regarding the need of a good clustering for outlier detection also the hybrid clustering is proposed (He *et al*., 2003; Jiang *et al*., 2001). Indeed, there is various outlier factor computation based on the used clustering algorithm within CBOD. Basically, the cluster and its structure reflect the pattern of the dataset therefore the deviation pattern can be used to detect outlier (Jiang *et al*., 2001; Duan *et al*., 2009). Shortly, the pattern deviation of particular data from common data within of the dataset is the most important aspect when outlier factor computation is developed. The use of pattern as the main foundation of outlier factor computation had been developed by some researchers and the development can be performed by clustering or supervised classification. Due to the advantage of using supervised approach on pattern classification, outlier factor measure is developed based on supervised classification (Chandola *et al*., 2009). For example, He *et al*. (2003) investigated the common pattern within data set by using association rule algorithm namely frequent item sets. Those infrequent patterns means that the rule only associated to small data and it can be categorized as an outlier. The aim of the use supervised outlier detection promises the better performance by incorporating specific knowledge about data set when outlier analysis is performed. Moreover, the supervised outlier detection is able to avoid the problem of high dimensional data processing.

Currently, Rough Set Theory is introduced in the outlier approach due to incomplete and uncertain dataset problem which has impacts on the sparse of the objects so the algorithm will be difficult to find outlier. Indeed Rough set Theory, proposed by Pawlak *et al*. (1995), is on based on the indiscernibility which highly accepted paradigm used to solve that problem. (Nguyen, 2007), deploys Granular computing and RST as a framework to elicit the knowledge which will be used to detect outlier. The use granular computing to calculate the object outlier factor is demonstrated Chen *et al*. (2008) successfully. The Degree outlierness for every object as the foundation to detect outlier based on RST also developed by Jiang *et al*. (2005) and Nguyen (2007). More recently, Shaari *et al*. (2009) proposed RSetOF to detect outlier successfully. RSetOF, which is computed using No Reduct, be able to maintain the performance although when it is used to detect in high dimensional dataset. However, in the real world, mostly the available data is unlabeled and lack of the knowledge. Furthermore, supervised outlier detection based is difficult to be implemented. Using the main problem which makes the use of RST on outlier detection, we propose to incorporate the clustering algorithm, indiscernibility and entropy to develop algorithm outlier detection.

Due to maintain the advantage of unsupervised classification when algorithm should deal with pattern generation, we develop suitable clustering and appropriate outlier factor computation. The main consideration of solving high dimension the dataset and uncertain data set problem is about:

- The loosely of the object within cluster
- The belonging of objects within cluster are more vague and complicated (Zhang *et al*., 2009)

Therefore we develop and extend the clustering algorithm which is addressed to deal that problem viz Rough K-Means Clustering (Lingras and West, 2004) as a foundation of our CBOD. The next section will describe the preliminaries of our study.

PROBLEM DEFINITION

In real-world datasets, scattered objects are common due to the high dimensionality and sparse feature value range problem. For example, we transform the Wisconsin (Diagnostic Breast Cancer) dataset into a two-dimensional projection as shown Fig. 1. Benign diagnoses are denoted by green points while malignant diagnoses are denoted by red triangles. The scattered normal (green) objects constitute a certain number of loosely bounded mini-clusters. Thus, by using the smaller cluster the outliers can be isolated.

In the case of a scattered dataset, we can assume that it is composed of many small clusters although in the real world, a dataset is just composed of a few clusters. Moreover, small clusters within scattered data have two important characteristics:

- Objects or/in small clusters are heavily distributed
- There is no precise boundary for a small cluster (Zhang *et al*., 2009)

Consequently, the size of cluster impacts on the other small clusters. If the amount of a small cluster is many, two or more small clusters are taken into consideration. The neighbourhood becomes sparse as more and more objects which belong to different small clusters should be taken into account. According to these characteristics, it is more reasonable to measure how the similarity among objects or clusters which is considered as the most dissimilar object or cluster as the outlier measurement. The most dissimilar cluster can be interpreted as the most separated clusters and this interpretation is valid on the objects of the most separated cluster. Referring to the classical outlier definition, we propose the use of an EBOF to measure the degree of object deviation in order to detect outliers. The fundamental of the EBOF measure is entropy for representing dissimilarity of the objects/clusters and the formal definition of the EBOF is introduced in the following section.

Cluster-based outliers detection: Generally, a cluster-based outlier is detected based on the amount of objects within the cluster. Thus, it is reasonable to suppose that any objects that do not lie in any large clusters and any objects that are not located in any area with close cluster are in fact outliers. If the clusters contain a small portion of objects then these clusters are candidates for cluster-based outliers (He *et al*., 2003; Duan *et al*., 2009). However, a problem arises since we cannot provide a definite number of objects that can be used as guidance to detect candidates for cluster-based outliers due to the variation of scattered real-world datasets. It is easier to develop a Cluster-Based Outlier Detection (CBOD) method by adopting the deviation concept of the outlier factor into cluster-based outlier detection. Hence, we give a definition of cluster-based outliers based on the deviation concept of the OF and conduct a detailed analysis.

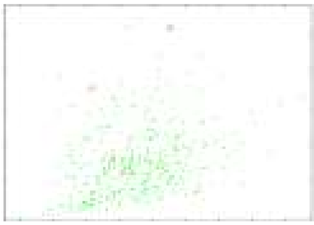

Fig. 1: Two-dimensional projection of Wisconsin dataset

The following shows how we can define cluster-based outliers and how our definition of the EBOF captures the spirit of the Hawkins definition. The fundamental of cluster-based outliers is that the higher deviation of a cluster the greater the possibility that there exist cluster-based outliers. The objects within the more possibility outlier cluster are the more possibility there is of there being outliers present. Suppose $C = \{C_1, C ..., C_n\}$ is a set of clusters which have sizes that are alike. Adopting the concept of the objects deviation OF into conventional CBOD; a cluster-based outlier is the cluster that is considerably deviates from the rest of the cluster in the entire dataset. Based on the foregoing, definition 1 can be formulated.

Definition 1: Suppose $C = \{C_1, C ..., C_n\}$ is a set of clusters, where $|C_1| \approx |C_2| \approx ... \approx |C_n|$ and $dif_C = \{dif_{C_1}, dif_{C_2}, ..., dif_{C_3}\}$ is the set of deviations for each cluster from the remaining clusters. Here, CBOD relies on the deviation value and it can be derived by using the properties of $dif_{C_1} \le dif_{C_2} \le dif_{C_3} \le .. \le dif_{C_{n-1}} \le dif_{C_n}$. Given the numeric parameter β, we can define C_o as an outlier cluster candidate:

$$C_o = \left\{ \underset{C_o \in C}{C_o} \; is\ candidate outlier cluster \,\middle|\, diff_{C_o} > dif_{Cb} \; and \; \left(dif_{Cb} / dif_{Cb-1} \ge \beta \right) \right\} \quad (1)$$

where,

$$dif_{Cb} / dif_{Cb-1} \ge \beta \quad (2)$$

The above reflects that the estimation of the outlier object can be limited by the characteristics of a cluster within the set of clusters. Here, the overlap clustering algorithm is required to produce clusters that are approximately equal in size. This approach gives an advantage in that we do not need to define the number of clusters precisely. We can estimate the number of clusters as approximately equal with the ratio of the number of outliers. For example, if we have a cluster to outlier ratio of approximately 1:10, then we can choose a number of partitions that is approximately equal to 10.

Parameter β in definition 1 gives a quantitative measure to distinguish among the cluster with a common class and the outlier clusters. Equation (1) considers the fact that most objects in the dataset are not outliers. Therefore, clusters that have a small deviation should be regarded as clusters with no outliers. Equation (2) considers that the quantitative deviation should have a significant value with respect to the outlier detection process.

Definition 2: Suppose $C_k = \{x_1, x_2 \ldots, x_n\}$ and $C_k \in C_o$. It is common that objects within a cluster are spread within the boundary of the cluster. Thus the dissimilarity measure of objects within outlier cluster into any other cluster is also varies. Considering this condition, an object within an outlier cluster is viewed as an outlier object candidate to be examined in the OF computation:

$$x = \left\{ \begin{array}{l} x \text{ } is \text{ } outlier \quad candidate \quad | \; x \in C_k \; and \\ diff_{C_k} = \min\left(dif_{C_1}, dif_{C_2}, \ldots, dif_{C_n}\right) \end{array} \right\} \tag{3}$$

Equation (3) is useful in that it creates a boundary area where the degree of deviation of objects within it should be calculated. Furthermore, the degree of deviation is used as a foundation for OF computation. Definition 2 and Eq. (3) represent deviation cluster provide important foundation for outlier detection. Thus the quality of the cluster significantly influences outlier detection. In the next subsection we will describe how to produce a cluster and deviation measurement.

Overlap clustering for cluster-based outliers detection: The existence of a cluster is a basic requirement to compute the OF. We extend Rough K-Means (RKM) (Lingras and West, 2004) to produce good overlap clustering. Rough K-Means is an interval clustering method which is powerful enough to deal with vague data processing. Figure 2 shows a set of interval cluster which represent upper approximation area. This set is divided into two parts, i.e., the Lower Approximation and Boundary Area. Since the objects' characteristics in the boundary area are not clear enough to be used in an OF computation we extend RKM as an overlap clustering algorithm.

Rough K-Means clustering (Lingras and West, 2004) produces a set of crisp clusters C, which is $C = \{\underline{C}_1, \ldots, \underline{C}_n\}$ and the boundary area B is $B = D - \underline{C}_1, \ldots, \underline{C}_n \cdot \; \underline{C}_1$. Whether or not to assign object x_l to the crisp cluster $\underline{appr}(c_j)$ or to its boundary area $\underline{bnd}(c_j)$ is calculated by Rough K-Means Algorithm (Lingras and West, 2004). This equation of RKM, which has been elaborated in Chapter IV, is used to assign membership and determine centroid reposition.

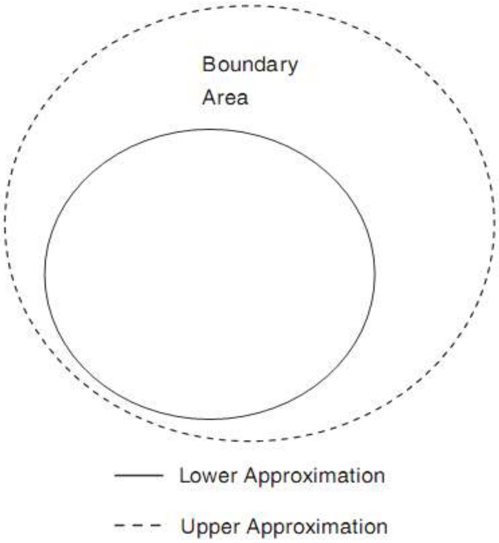

Fig. 2: Lower and upper approximation in cluster C

Definition 3: The membership (of any object within the boundary area $B = D - \underline{C}_1, \ldots, \underline{C}_n$ can be used as an assignment foundation into the suitable crisp cluster $C = \{\underline{C}_1, \ldots, \underline{C}_n\}$. The value can be interpreted as the degree that x belongs to $\underline{C}_i$ in view of the knowledge expressed about x. Furthermore, by using the concept of the discernibility of Rough Set Theory (RST) approach, is computed as in Eq. (4) below:

$$\mu_x^{c_i}(x) = \frac{\left|\left[x_c^*\right] \cap c_i\right|}{\left|\left[x_c^*\right]\right|} \tag{4}$$

Definition 3: Is one of the important characteristics in RST, i.e., the approximation computation. It is used to assign vague objects within the boundary area by using the rough membership concept of RST. It improves the performance of the clustering algorithm, especially when the algorithm should deal with small clusters which are used to compute the OF. In the next subsection we will introduce entropy as the foundation of the OF computation.

Entropy-based outlier factor computation: Entropy in information theory is associated with the orderly or disorderly configuration of data. Using a reasoning concept, a disorderly configuration can be interpreted as denoting that most of the data points are scattered randomly. Shannon's entropy explains that when the probability of mass function is uniformly distributed, the degree of ‘disorder’ is higher. A disorderly configuration measures the entropy which represents some “degree of similarity or deviation” of the data (Yao and Dash, 2000).

Definition 4: Suppose *i* and *j* are two points within a dataset, then the entropy can be calculated as follows:

$$E_{ij} = -(S_{ij} \log_2 S_{ij} + (1 - S_{ij}) \log_2 (1 - S_{ij})) \quad (5)$$

where S_{ij} is the similarity between point *i* and point *j*. Moreover, the similarity is based on the distance measurement. E_{ij} assumes the maximum value of 1 when S_{ij} is 0.5, which means that the distance of this point is close to the mean distance of all pairs within the dataset. E_{ij} will have a minimum value (close to 0.0) if the distance is very close or very distant.

PROPOSED ALGORITHM

We develop an outlier detection algorithm based on the cluster outliers and this entropy. Given the set of clusters $C = \{C_1,\dots, C_n\}$, according to Eq. (3), cluster C_x and its objects are outlier candidates if this cluster C_x has some characteristics that differ greatly from those of other clusters in $C = \{C_1,\dots, C_n\}$. The object in the most different cluster is considered as an outlier cluster candidate. The degree of deviation for each object within the outlier cluster is calculated based on the entropy similarity concept (Yao and Dash, 2000).

Definition 5: Let $C = \{C_1,\dots, C_n\}$ be a set of clusters which is generated as $C_i \approx C_j \Leftrightarrow c_i = c_j$, where c_i, c_j are the centroid of the clusters *i* and *j*. c_i, c_j are used to measure entropy which represents the similarity measurement among the clusters:

$$E_i = -\sum_{j \in S}^{j \neq i} (D_{ij} \log_2 D_{ij} + (1 - D_{ij}) \log_2 (1 - D_{ij})) \quad (6)$$

where, $D_{ij} = e^{-\alpha \sqrt{\|c_i - c_j\|^2}}$ and c_i = centroid cluster *i*, c_j = centroid cluster *j*.

D_{ij} is the similarity between c_i and cc_j normalized to [0.0-l.O]. In the c-means clustering framework, objects are assigned based on their similarity to the centroid and each object within the same cluster is assumed have similar properties. Definition 5 represents the properties of c-means clustering framework which map objects into their centroid. This definition is useful as it can reduce the computation load when the entropy computation is performed.

Definition 6: Suppose $E = \{E_1, E_2, \dots, E_n\}$ is the set of entropies of the set of cluster $C = \{C_1,\dots, C_n\}$ and $|E_1| \le |E_2| \le \dots \le |E_n|$. The sequence of entropy represents the property of the cluster in the dataset. The minimum entropy among the set of cluster *C* can be used to detect whether this cluster is a normal cluster or outlier cluster. Given the numeric parameter *β*, according to Eq. (1) the outlier candidate can be detected as follows:

$$C_o = \left\{ \underset{C_o \in C}{C_o} \; is\ candidate\ outlier\ cluster \;\middle|\; E_o > E_n \ and \ \left(E_n / E_{n-1} \ge \beta\right) \right\} \quad (7)$$

Parameter *β* is used to ensure that there is a significant difference between common clusters and an outlier cluster.

Definition 7: Let $C_o = \{x_1, \dots, x_n\}$ and E_o be the entropy of cluster C_o. All of $x_n \in C_o$ are outlier object candidates that need to be examined using entropy. The entropy is calculated based on the outlier object candidates and common clusters. Moreover, this entropy is used as the OF objects x_n. Referring to the properties of clusters, all of cluster centroid is used as a point to measure the OF of objects x_n:

$$EBOF_{x_n} = OF_{x_n} \ * \ E_o \quad (8)$$

where,

$$OF_{x_n} = -\sum_{C_k \in C \ and \ C_k \neq C_o} (D_{C_k x c_n} \log_2 D_{C_k x c_n} + (1 - D_{C_k x c_n}) \log_2 (1 - D_{C_k x c_n})) \quad (9)$$

and,

$$D_{C_k x c_n} = \sqrt{\|x_n - c_k\|^2}$$

Equation (8) and (9) are derived from the entropy characteristics. Thus we can transform the entropy as the OF as formulated in Eq. (8). Furthermore, the OF is ranked based on the aim of outlier detection. To summarize, we describe CBOD using an entropy computation as follows.

Entropy calculation for outlier detection:

Step 1: Calculate the entropy of the entire cluster using Eq. (6)
Step 2: Build the sequence of the outlier clusters based on the entropy
Step 3: Using a cut-off point chooses the candidate outlier clusters as in Eq. (7)

Step 4: Calculate the entropy objects in all of the outlier clusters using Eq. (8) and (9)
Step 5: Build the sequence of objects based on the entropy value
Step 7: Detect and rank the outlier objects based on the entropy value

IMPLEMENTATION AND DISCUSSION

A comprehensive performance study has been conducted to evaluate our algorithm. In this section, we present the results of the analysis and testing of our algorithm using datasets from the UCI Machine Learning Repository. Several experimental tests were executed then the result was compared with the results for algorithm that have a similar purpose, i.e., the greedy algorithm (He *et al*., 2006), the Find FPOF (Frequent Pattern Outlier Factor) (He *et al*., 2004) and the RSetOF (Shaari *et al*., 2009). Furthermore, to measure the performance, we use the top ratio and coverage ratio. The top ratio is the ratio of the number of records specified as top-k outliers to that of the records in the dataset while the coverage ratio is the ratio of the number of detected rare classes to that of the rare classes in the dataset. Since the clustering algorithm uses random initialization to start the process, we repeat each experiment 50 times. The average outcome of the process is then compared with the other algorithms. The detail of our experiment is described as below.

Time complexity analysis: Given dataset D with a condition of an amount of records n, the number of attributes is m while the number of outliers is k

Greedy algorithm: The foundation of the greedy algorithm is entropy which needs at most $O\ (m*p)$ (He *et al*., 2006), therefore the complexity is $O\ (n*k*m)$.

Find FPOF: There are three steps in Find FPOF (He, 2005), i.e.:

- Mining the frequent patterns with complexity $O = (FP)$
- Calculating Find FPOF complexity $O = (N*S)$
- Finding using the sorting approach N*log N + S* $(top\text{-}n)*top\text{-}k)$*log $(top\text{-}k)$ and the total of complexity is $O = (N*S) + N$*log $N + S* (top\text{-}n)*top\text{-}k)$*log $(top\text{-}k)$

RsetOF: There are three steps in RsetOF (Shaari *et al*., 2009), i.e.:

- Extracting rule from decision tables by using Genetic Algorithm. time complexity is $O = n^2 *m$
- Calculating the degree of support as RsetOF
- Sorting RsetOF to detect outliers, where time complexity is N*log N and the total complexity is $O\ (n^2*m+N$*log $N)$

EBOF: The steps of the proposed EBOF also number three as follows:

- RKM clustering is essentially equal to K-Means clustering. The difference between them is that in RKM clustering, the process is performed two times, i.e., clustering the lower approximation and clustering the boundary area (see this Section). The time complexity is $O\ (n*c*I*m)$, where c is the number of clusters and I is the number of iterations.
- Calculating and sorting entropy among the centroids of cluster, where the time complexity is $O\ (c*\log c)$.
- Calculating and sorting the entropy between an object within an outlier cluster and another centroid of a clusters, where the time complexity is:

$$O((c-1)*|C_{outlier}|*\log(c-1)*|C_{outlier}|)$$

Based on the complexity analysis we can predict how big the computation cost will be. For example, the greedy algorithm and a simple algorithm which uses the entropy concept are comparable in terms of computation cost. Furthermore, using RKM, which is an extension of c-means clustering framework and entropy which is developed based on distance similarity, the simplicity and complexity still can be maintained. However, if we compare our proposed method with RsetOF, the complexity achieved by our algorithm is better, since RsetOF uses an Evolutionary Algorithm (EA). This is because the EA is a well-known global optimization method while RKM is performed with aim of achieving local minima optimization using random initialization.

Experiment and results: Our outlier detection method is based on RKM, indiscernibility and entropy, the aim of which is to improve the performance of outlier detection and to the reduce time complexity problem. The first experiment is oriented to demonstrate the ability of the proposed outlier detection algorithm on a real-world dataset which has just a few overlap among the classes and a small dimension, i.e., the UCI Iris dataset. The second and third experiment are intended to compare our results with other outlier detection methods (which are also rough set based) on datasets that have a high dimension and a small overlap between classes within the dataset domain, i.e., the UCI repository datasets, Glass and Wisconsin. The fourth experiment is designed to compare the performance of our algorithm with other outlier detection methods (which are not rough set based), where the algorithm

Table 1: Class distribution of iris dataset

Case	Class codes	Percentage of instances
Commonly occurring classes	2, 3	92.23
Rare classes	1	7.77

should deal with high-dimensional data and special characteristics such as identifying that the positive examples belong to a class and the negative examples belong to a different class and for this purpose we test our algorithm on the UCI the *E. coli* dataset. Furthermore, the aim of this experiment is to show the performance of the proposed algorithm in relation to the described problem.

Iris dataset: The Iris dataset is used since this dataset is a well-known database in the pattern recognition literature. The dataset contains three classes of 50 instances each, where each class refers to a type of iris plant. Another important characteristic of the Iris dataset is that it only has one class which is linearly separable from the other two. In this first experiment, some instances of the Iris dataset were removed randomly to form an imbalanced distribution. A number of (3) instances were removed from the class code 0, while (2) instances was removed from the class code 2 and (42) instances were removed from the class code 1. As a result, the class codes 0 and 2 contained (95) instances or 92.23%, which can be referred to as the common class, whereas class code 1 contained (8) instances or 7.77%, which can be referred to as the rare class (Table 1).

The results in Table 2 below show that, on average, the proposed algorithm demonstrated the best performance to detect outliers. Firstly, it showed better performance in its outlier detection rate, which is represented by the lower value of the top ratio 10.68% (11). Note that the lower the outlier detection rate is the higher the speed of outlier detection is. Secondly, in this experiment the characteristic of the increment of outlier detection tends to be constant since the outlier is detected as the cluster which has smallest entropy in average.

Wisconsin breast cancer data: The second dataset is the Wisconsin breast cancer dataset, which has 699 instances with nine attributes and each record is labelled as benign (458; 65.5%) or malignant (241; 34.5%). We follow the experimental technique of Harkins *et al*. (2002) by removing some of the malignant records to form a very unbalanced distribution; the resultant dataset had 39 (8%) malignant records and 444 (92%) benign records (Table 3).

From Table 4, it can be seen that for the Wisconsin breast cancer dataset, the RSetOF was the best followed by EBOF. This condition is supported by the fact that the RSetOF is able to detect all the rare cases in the top 39 ranked records faster than EBOF. This might have happened because the amount of sparse and overlapping data forces another algorithm to become trapped within local minima during optimization. However, this better performance is offset by the high cost computation of the RSetOF algorithm.

Glass: The Glass dataset used in the experiment originally contained 214 instances with 10 attributes, but we removed (1) from a total of 11 attributes: the ID attribute was removed as it did not have any significant impact on the dataset. The class codes 1, 2, 3, 5 and 7 were grouped as commonly occurring classes with a larger number of instances (95.77%). The remaining class code 6 was used as the rare class with small number of instances (4.23%) (Table 5).

From Table 6, it can be seen that although the proposed algorithm cannot detect the first (0.47%) and the second (0.94%) outlier, this proposed method still was able to outperform the other algorithms. While it

Table 2: Detected outliers in iris dataset

	Number of outliers belonging to rare class (coverage ratio in %)			
Top ratio in %	FindFPOF	RSetOF	GreedyAlg	Proposed algorithm EBOF
0.97% (1)	1 (12.50%)	0%	1 (12.50%)	0%
1.94% (2)	2 (25.00%)	0%	2 (25.00%)	0%
2.91% (3)	2 (25.00%)	1 (12.50%)	2 (25.00%)	1 (12.50%)
3.88% (4)	3 (37.50%)	2 (25.00%)	2 (25.00%)	2 (25.00%)
4.85% (5)	4 (50.00%)	2 (25.00%)	2 (25.00%)	3 (37.50%)
5.83% (6)	5 (62.50%)	2 (25.00%)	2 (25.00%)	4 (50.00%)
6.80% (7)	6 (75.00%)	2 (25.00%)	2 (25.00%)	4 (50.00%)
7.77% (8)	7 (87.50%)	2 (25.00%)	3 (37.50%)	5 (62.50%)
8.74% (9)	7 (87.50%)	3 (37.50%)	4 (50.00%)	6 (75.00%)
9.71% (10)	7 (87.50%)	4 (50.00%)	5 (62.50%)	7 (87.50%)
10.68% (11)	7 (87.50%)	5 (62.50%)	6 (75.00%)	8 (100.00%)
11.65% (12)	8 (100.00%)	6 (75.00%)	7 (87.50%)	8 (100.00%)
12.62% (13)	8 (100.00%)	6 (75.00%)	8 (100.00%)	8 (100.00%)
13.59% (14)	8 (100.00%)	7 (87.50%)	8 (100.00%)	8 (100.00%)
14.59% (15)	8 (100.00%)	8 (100%)	8 (100.00%)	8 (100.00%)
15.53% (16)	8 (100.00%)	8 (100%)	8 (100.00%)	8 (100.00%)

Statistics of EBOF: Best: 6.80% (7); Average: 10.68% (11); Worst: 26.21% (27)

Table 3: Class distribution of Wisconsin dataset

Case	Class codes	Percentage of instances
Commonly occurring classes	2	92
Rare classes	4	8

showed the best performance (4.23% (9)) and the worst performance (22.00% (49)) its coverage ratio as still lower than the other comparator algorithms. Therefore, this proposed method, which has the capability to detect all outliers with 100% coverage ratio at the lowest top ratio can be considered to have a very high detection rate. That is to say, the proposed algorithm is capable of detecting outliers more effectively than the other two algorithms.

Table 4: Detected outliers in Wisconsin dataset

	Number of outliers belonging to rare class (coverage ratio in %)			
Top ratio in %	FindFPOF	RSetOF	GreedyAlg	Prop. algorithm EBOF
0% (0)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
1% (4)	3 (7.69%)	5 (12.82%)	4 (10.26%)	4 (10.26%)
2% (8)	7 (17.95%)	8 (20.51%)	7 (17.95%)	8 (20.51%)
4% (16)	14 (35.90%)	19 (48.72%)	15 (38.46%)	16 (41.02%)
6% (24)	21 (53.85%)	28 (71.79%)	22 (56.41%)	24 (61.53)
8% (32)	28 (71.79%)	39 (100.00%)	27 (69.23%)	29 (74.35)
10% (40)	31 (79.49%)	39 (100.00%)	33 (84.62%)	39 (100.00%)
12% (48)	35 (89.74%)	39 (100.00%)	36 (92.31%)	39 (100.00%)
14% (56)	39 (100.00%)	39 (100.00%)	39 (100.00%)	39 (100.00%)

Statistics of EBOF: Best: 15.53% (39); Average: 17.46% (44); Worst: 37.30% (94)

Table 5: Class distribution of glass dataset

Case	Class codes	Percentage of instances
Commonly occurring classes	1, 2, 3, 5 and 7	95.77
Rare classes	6	4.23

Table 6: Detected outliers in glass dataset

	Number of outliers belonging to rare class (coverage ratio in %)			
Top ratio in %	FindFPOF	RSetOF	GreedyAlg	Prop. alg. EBOF
0.47% (1)	0 (0%)	1 (11.11%)	0 (0%)	0 (0%)
1.41% (3)	0 (0%)	2 (22.22%)	0 (0%)	1 (11.11%)
2.82% (6)	0 (0%)	2 (22.22%)	0 (0%)	2 (22.22%)
3.29% (7)	1 (11.11%)	2 (22.22%)	1 (11.11%)	4 (44.44%)
3.76% (8)	1 (11.11%)	2 (22.22%)	2 (22.22%)	5 (55.56%)
4.23% (9)	1 (11.11%)	2 (22.22%)	2 (22.22%)	6 (66.67%)
6.10% (13)	2 (22.22%)	2 (22.22%)	2 (22.22%)	9 (100.00%)
6.57% (14)	3 (33.33%)	2 (22.22%)	2 (22.22%)	9 (100.00%)
7.04% (15)	3 (33.33%)	2 (22.22%)	2 (22.22%)	9 (100.00%)
8.03% (81)	3 (33.33%)	2 (22.22%)	2 (22.22%)	9 (100.00%)
0.38% (86)	4 (44.44%)	2 (22.22%)	2 (22.22%)	9 (100.00%)
0.85% (87)	5 (55.56%)	2 (22.22%)	2 (22.22%)	9 (100.00%)
1.31% (88)	5 (55.56%)	2 (22.22%)	2 (22.22%)	9 (100.00%)
3.19% (92)	5 (55.56%)	3 (33.33%)	3 (33.33%)	9 (100.00%)
3.66% (93)	5 (55.56%)	4 (44.44%)	4 (44.44%)	9 (100.00%)
4.13% (94)	5 (55.56%)	5 (55.56%)	5 (55.56%)	9 (100.00%)
4.60% (95)	5 (55.56%)	6 (66.67%)	6 (66.67%)	9 (100.00%)
7.42% (101)	6 (66.67%)	6 (66.67%)	6 (66.67%)	9 (100.00%)
7.89% (102)	7 (77.78%)	6 (66.67%)	6 (66.67%)	9 (100.00%)
6.34% (120)	7 (77.78%)	6 (66.67%)	6 (66.67%)	9 (100.00%)
6.20% (141)	7 (77.78%)	9 (100.00%)	6 (66.67%)	9 (100.00%)
7.14% (143)	7 (77.78%)	9 (100.00%)	6 (66.67%)	9 (100.00%)
3.57% (155)	8 (88.89%)	9 (100.00%)	6 (66.67%)	9 (100.00%)
4.04% (156)	9 (100.00%)	9 (100.00%)	6 (66.67%)	9 (100.00%)
6.38% (157)	9 (100.00%)	9 (100.00%)	7 (77.78%)	9 (100.00%)
6.24% (205)	9 (100.00%)	9 (100.00%)	7 (77.78%)	9 (100.00%)
96.71% (206)	9 (100.00%)	9 (100.00%)	8 (88.89%)	9 (100.00%)
97.18% (207)	9 (100.00%)	9 (100.00%)	9 (100.00%)	9 (100.00%)

Statistics of EBOF: Best: 3.73% (8); Average: 6.07% (13): Worst: 18.22% (39)

Table 7: Class distribution of *E. coli* dataset

Case	Class codes	Percentage of instances
Commonly occurring classes	1, 2, 5, 6 and 8	97.31
Rare classes	3, 4 and 7	2.69

Table 8: Detected outliers in *E. coli* dataset

	Number of outliers belonging to rare class (coverage ratio in %)			
Top ratio in (%)	FindFPOF	RSetOF	GreedyAlg	Prop. alg. EBOF
0.30	1 (11.11%)	1 (11.11%)	1 (11.11%)	1 (11.11%)
0.60	1 (11.11%)	2 (22.22%)	2 (22.22%)	2 (22.22%)
0.90	2 (22.22%)	3 (33.33%)	2 (22.22%)	2 (22.22%)
1.19	3 (33.33%)	4 (44.44%)	2 (22.22%)	2 (22.22%)
1.49	3 (33.33%)	5 (55.56%)	3 (33.33%)	3 (33.33%)
1.79	4 (44.44%)	6 (66.67%)	4 (44.44%)	3 (33.33%)
2.09	5 (55.56%)	7 (77.78%)	5 (55.56%)	4 (44.44%)
2.39	6 (66.67%)	7 (77.78%)	5 (55.56%)	5 (55.56%)
2.69	7 (77.78%)	7 (77.78%)	6 (66.67%)	5 (55.56%)
3.88	7 (77.78%)	7 (77.78%)	7 (77.78%)	6 (66.67%)
3.28	7 (77.78%)	7 (77.78%)	7 (77.78%)	7 (77.78%)
3.58	8 (88.89%)	7 (77.78%)	7 (77.78%)	7 (77.78%)
3.88	8 (88.89%)	7 (77.78%)	7 (77.78%)	7 (77.78%)
4.18	8 (88.89%)	7 (77.78%)	7 (77.78%)	7 (77.78%)
4.48	9 (100.00%)	7 (77.78%)	7 (77.78%)	7 (77.78%)
4.78	9 (100.00%)	7 (77.78%)	8 (88.89%)	8 (88.89%)
14.93	9 (100.00%)	7 (77.78%)	9 (100.00%)	9 (100.00%)
29.85	9 (100.00%)	7 (77.78%)	9 (100.00%)	9 (100.00%)
44.78	9 (100.00%)	7 (77.78%)	9 (100.00%)	9 (100.00%)
66.27	9 (100.00%)	8 (88.89%)	9 (100.00%)	9 (100.00%)
66.57	9 (100.00%)	9 (100.00%)	9 (100.00%)	9 (100.00%)

Statistics of EBOF: Best: 3.60% (8); Average: 10.81% (24); Worst: 24.32% (54)

***E. coli*:** The *E. coli* dataset originally contained 335 instances with 8 attributes including the decision attribute. The class codes 1, 2, 5, 6 and 8 were grouped as commonly occurring classes with a larger number of instances (97.31%). The remaining class codes 3, 4 and 7 contained a small number of instances, equivalent to 2.69% and were regarded as the rare class (Table 7).

The results in Table 8 show that in this experiment the best algorithm was Find FPOF, while EBOF was comparable with the greedy algorithm. The greedy algorithm uses entropy as a foundation computation. The difference between the greedy algorithm is that EBOF relies on the centroid of the cluster as a foundation to compute the OF. So, by improving the quality of the cluster, the performance will increase.

Discussion: From the four experiments above, compared with similar algorithms which are designed to detect outliers within high-dimensional and uncertain datasets, the proposed EBOF algorithm performed the best especially in processing a numeric dataset i.e., Iris and Glass dataset and its performance was comparable when run on a mixed dataset (numeric and categorical) i.e., Wisconsin and *E. coli* datasets. With respect to the distance as a foundation of similarity or proximity approach, this algorithm was notable to solve sparse or uncertain dataset in terms of the outlier detection.

According to the result of the experiment, compared to previous algorithms, the proposed algorithm has some important advantages:

- The scatter data problem due to the curse of high dimensionality is solved by the use of overlap clustering and entropy measure within the centroid cluster. The use of clustering in high-dimensional data is able to reduce complexity by mapping objects into a small cluster. Furthermore, the cluster is able to represent the data for the next processing viz. entropy computation. The use of the centroid cluster as a cluster representation avoids the uncertainty problem that occurs in high-dimensional datasets. This is because the algorithm uses the appropriate cluster to produce the appropriate centroid which is used in the next computation. Hence the scatter problem in the high-dimensional computation is reduced. This phenomenon was clearly apparent in the Glass dataset, where EBOF outperformed the other algorithms.
- The use of the indiscernibility of RST in association with clustering is an extension of RKM that also improves the capability of interval clustering. This improvement occurs as a result of using the approximation concept of RST to deal with vague data in the boundary area within the clusters.
- The efficacy of using similarity based on an entropy calculation has been proved when it was applied in clustering; however, this approach has the impact of a high cost computation (Yao and Dash, 2000). Nevertheless, we are able to use this concept to our advantage by measuring the dissimilarity to detect outlier objects and the result is that the EBOF has a high detection rate while the computation cost is maintained by using the centroid as the basis for computation in our algorithm.

In summary, the above experimental results showed that the EBOF can outperform previous algorithms in the case of two datasets (Iris and Glass) and its performance is comparable in the case of one dataset (Wisconsin). Moreover, in relation to the complexity analysis, EBOF has no significant additional computation cost when compared with RSetOF which uses a EA for optimization. This means that the EBOF algorithm can discover outliers more efficiently and effectively than the other three algorithms. However, according to the statistical result (minimum, average and maximum) the performance of the EBOF algorithm might be enhanced further by using an optimization of c-means clustering framework in order to improve its outlier detection rate.

CONCLUSION

The existing outlier detection schemes are either distance-based or density-based. Their capabilities of outlier detection are based on methods that mostly using the distance measurement as a foundation of outlier detection. We have proposed a concept of measuring dissimilarity as a basis of outlier detection that can be performed on various dimensional dataset. Based on our experiment, the capabilities of entropy, RKM and indiscernibility enable enhancement of the distance-based computation approach to detect outliers. When we compared our approach with previous algorithms, we have found that the EBOF is more effective in an environment where patterns are numerical in datasets which have both a low and a high ratio of outliers, while it is comparable in the detection outliers in a mixed dataset. However, performance might be improved since the optimization of overlap clustering, which is an important component, has not yet been performed especially on the mixed dataset.

Another interesting issue that has been identified relates to the refinement of outlier detection. The proposed algorithm which is aimed to refine outlier detection is based on the dissimilarity distance, which is converted to an entropy measurement. This approach enables distance-based algorithm to solve the sparse data problem. The use of indiscernibility improves the ability of K-Means framework clustering to distinguish vague objects. Our study on the capabilities of the CBOD approach has shown that the use of appropriate clustering and the classical concept (distance-based approach for outlier detection) is still relevant for outlier detection. However, due to the disadvantage of the distance-based approach, our algorithm should be developed to enhance its effectiveness for categorical datasets. Therefore, further study is needed with respect to the application of our algorithm to categorical datasets.

ACKNOWLEDGMENT

I would like to acknowledge Atma Jaya University in Yogyakarta, Indonesia and UKM Grant No.UKM-DLP-2011-020 for the financial support of this study project.

REFERENCES

Breunig, M.M., H.P. Kriegel, R.T. Ng and J. Sander, 2000. LOF: Identifying density based local outliers. Proceedings of the ACM SIGMOD International Conference on Management of Data, pp: 93-104.

Chandola, V., A. Banerjee and V. Kumar, 2009. Anomaly detection: A survey. ACM Comput. Surv., 41(3), Article 15.

Chen, Y., D. Miao and R. Wang, 2008. Outlier Detection Based on Granular Computing. Springer, Heidelberg.

Duan, L., L. Xu, Y. Liu and J. Lee, 2009. Cluster-based outlier detection. Ann. Oper. Res., 168: 151-168.

Harkins, S., H. He, G.J. Willams and R.A. Baster, 2002. Outlier detection using replicator neural networks. Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery. Aix-en-Provence, France, pp: 170-180.

Hawkins, D.M., 1980. Identifications of Outliers, Monograph on Applied Probability and Statistic. Chapman and Hall, London.

He, Z., X. Xu and S. Deng, 2003. Discovering cluster based local outliers. Pattern Recogn. Lett., 24(9-10): 1641-1650.

He, Z., S. Deng and X. Xu, 2005. An optimization model for outlier detection in categorical data. Proceeding of the International Conference on Intelligent Computing, pp: 400-495.

He, Z., J.Z. Huang, X. Xu and D. Shengchun, 2004. A Frequent Pattern Discovery Method for Outlier Detection. In: Springer Link (Ed.), Lecture Notes Computer Science. Springer, Berlin/Heidelberg, pp: 726-732.

He, Z., S. Deng, X. Xu and J.Z. Huang, 2006. A fast greedy algorithm for outlier mining. Proceeding of 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD, 2006), pp: 567-576.

Hodge, V.J. and J. Austin, 2004. A survey of outlier detection methodologies. Artif. Intell. Rev., 22: 85-126.

Jiang, M.F., S.S. Tseng and C.M. Su, 2001. Two-phase clustering process for outliers detection. Pattern Recogn. Lett., 22(6-7): 691-70.

Jiang, F., Y.F. Sui and C.G. Cao, 2005. Outlier Detection Using Rough Set Theory. In: Ślęzak, D., J. Yao, J.F. Peters, W. Ziarko and X. Hu (Eds.), RSFDGrC 2005. LNCS (LNAI), Springer, Heidelberg, 3642: 79-87.

Jiang, F., Y. Sui and C. Cao, 2006. Outlier Detection Based on Rough Membership Function. In: Greco S. *et al.* (Eds.), RSCTC 2006. LNAI 4259, Springer-Verlag, Berlin, Heidelberg, pp: 388-397.

Lingras, P. and C. West, 2004. Interval set clustering of Web users with rough k-means. J. Intell. Inform. Syst., 23: 5-16.

Mahoney, M.V. and P.K. Chan, 2003. Learning rules for anomaly detection of hostile network traffic. Proceeding of 3rd IEEE International Conference on Data Mining (ICDM, 2003), pp: 601-604.

Nguyen, T.T., 2007. Outlier Detection: An Approximate Reasoning Approach. In: Kryszkiwicz, M. *et al*. (Eds.), RSEISP 2007. LNAI 4585, Springer-Verlag, Berlin, Heidelberg, pp: 495-504.

Pawlak, Z., J. Grzymala-Busse, R. Slowinski and W. Ziarko, 1995. Rough sets. Commun. ACM, 38(11): 89-95.

Pires, A. and C.M. Santos-Pereira, 2005. Using clustering and robust estimators to detect outliers in multivariate data. Proceedings of the International Conference on Robust Statistics.

Ramaswamy, S., R. Rastogi and K. Shim, 2000. Efficient algorithms for mining outliers from large data sets. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD'00), pp: 427-438.

Shaari, F., A.A. Bakar and A.R. Hamdan, 2009. Outlier detection based on rough sets theory. Intell. Data Anal., 13(2): 191-206.

Yao, J. and M.M. Dash, 2000. Entropy-based fuzzy clustering and modeling. Fuzzy Set. Syst., 3: 282-188.

Zhang, K., M. Hutter and H. Jin, 2009. A new local distance-based outlier detection approach for scattered real-world data. Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD, 2009), pp: 813-822.