American Journal of Intelligent Systems 2012, 2(5): 129-140
DOI: 10.5923/j.ajis.20120205.07

RODHA: Robust Outlier Detection using Hybrid Approach

A. Mira*, D.K. Bhattacharyya, S. Saharia

Dept. of Computer Science & Engg., Tezpur University, Sonitpur, Assam, 784028, India

Abstract The task of outlier detection is to find the small groups of data objects that are exceptional to the inherent behavior of the rest of the data. Detection of such outliers is fundamental to a variety of database and analytic tasks such as fraud detection and customer migration. There are several approaches[10] of outlier detection employed in many study areas amongst which distance based and density based outlier detection techniques have gathered most attention of researchers. In information theory, entropy is a core concept that measures uncertainty about a stochastic event, and it means that entropy describes the distribution of an event. Because of its ability to describe the distribution of data, entropy has been applied in clustering applications in data mining. In this paper, we have developed a robust supervised outlier detection algorithm using hybrid approach (RODHA) which incorporates both the concept of distance and density along with entropy measure while determining an outlier. We have provided an empirical study of different existing outlier detection algorithms and established the effectiveness of the proposed RODHA in comparison to other outlier detection algorithms.

Keywords Distance Based , Density Based, Entropy, Locality Sensitive Outlier

1. Introduction

The majority of the earlier research works of data mining focussed on the general pattern applicable to the larger section of the data. On the other hand, outlier detection focuses on that smaller section of data that exhibit exceptional behaviour compared to the rest large amount of the data. A well-quoted definition of outliers is first given by Hawkins[12]. It states, "An outlier is an observation that deviates so much from other observations so as to arouse suspicion that it was generated by a different mechanism". Outlier detection, since its inception has been regarded as an important aspect for study in data mining research as it uncovers the valuable knowledge hidden behind whole data and aiding the decision makers to make profit or improve the service quality. Outlier detection has several applications. For example, outlier detection can be employed as a pre-processing step to clean the data set from erroneous measurements and noisy data points. On the other hand, it can also be used to isolate suspicious or interesting patterns in the data. Examples include fraud detection, customer relationship management, network intrusion, clinical diagnosis and biological data analysis.

In this paper we have provided an empirical study of some existing outlier detection techniques. We have done a detail theoretical study and implementation of Locality Sensitive Hashing (LSH)-based outlier detection technique proposed by Wang (et. al)[20]. Apart from this we have proposed a robust outlier detection algorithm using a hybrid approach (RODHA) based on both distance and density based approach along with incorporating the entropy measure to determine the outliers. The proposed RODHA can be found to be significant in view of the following points.

- Free from the restriction of the using specific proximity measure.
- Takes the benefit of distance based, density based as well as information theoretic approach while identifying an outlier.
- Sensitive and scalable.
- Performance is independent of dimensionality and number of clusters.

Rest of the paper is organized as follows: section 2 reports related research. In section 3, we provide the background of our work. In section 4, the LSH-based outlier detection technique is described in brief. Section 5 presents the proposed RODHA approach and the empirical evaluation of the method is reported in detail in section 6. Finally, concluding remarks and future direction of research is given in section 7.

2. Related Research

There are two kinds of outlier detection methods: formal tests and informal tests [22]. Formal and informal tests are usually called tests of discordance and outlier labelling

* Corresponding author:
arifuzzamanmira@gmail.com (A. Mira)

Published online at http://journal.sapub.org/ajis

methods, respectively.

Most formal tests need test statistics for hypothesis testing. They are usually based on assuming some well-behaving distribution, and test if the target extreme value is an outlier of the distribution, i.e., whether or not it deviates from the assumed distribution. Some tests are for a single outlier and others for multiple outliers. Selection of these tests mainly depends on numbers and type of target outliers and type of data distribution. Even though formal tests are quite powerful under well-behaving statistical assumptions such as a distribution assumption, most distributions of real-world data may be unknown or may not follow specific distributions such as the normal, gamma, or exponential. Another limitation is that they are susceptible to masking or swamping problems.

On the other hand, most outlier labelling methods, informal tests, generate an interval or criterion for outlier detection instead of hypothesis testing, and any observations beyond the interval or criterion is considered as an outlier. There are two reasons for using an outlier labelling method. One is to find possible outliers as a screening device before conducting a formal test. The other is to find the extreme values away from the majority of the data regardless of the distribution. Some very popular outlier labelling parameters are Z-score[22], Standard deviation (SD) method[22], Turkey's method, MAD_e method[22] and Median Rule[22].

In data mining, the problem of outlier detection has been tried to solve based on several approaches [10] in different problem domains. The class of solution to outlier detection ranges from statistical methods to geometric methods and from density based approaches to distance based approaches. Statistical methods are appropriate if one has a good sense for the background distribution but typically does not scale well to large datasets or datasets of even moderate dimensionality. Geometric methods essentially rely on variants of the convexhull algorithm which has a complexity that is exponential in the dimensionality of the data, and they are often impractical. The distance-based approach[15] originally proposed by Ng and Knorr. They define a point to be a distance-based outlier if at least a user-defined fraction of the points in the dataset are further away than some user-defined minimum distance from that point. In their experiments, they primarily focus on datasets containing only continuous attributes. This can be expensive to compute particularly in higher dimensions. A standard distance based approach called ORCA[3] proposed by Stephen D. Bay employs some pruning rule for optimization of processing time in large multi-dimensional datasets. Because of this pruning rule the algorithm scales well to a linear time in case of high dimensional dataset. Another distance based approach in conjunction with a ranking scheme is the Locality Sensitive Hashing (LSH)-based outlier detection proposed by Wang (et. al)[20]. Here the outlier ranking scheme is based on a hashing concept called Locality Sensitive Hashing (LSH). The basic idea for LSH is to convert the data into manageable fingerprints and hash them so that similar data points are mapped to the same buckets with high probability. Density-based approaches [4] to outlier detection rely on the computation of the local neighbourhood density of a point. In one such technique, a local outlier factor (LOF) is computed for each point. The LOF of a point is based on the ratio of the local density of the area around the point and the local densities of its neighbours. The size of a neighbourhood of a point is determined by the area containing a user-supplied minimum number of points (*MinPts*). Pang-Ning Tan proposed *OutRank-b*[16], a graph-based outlier detection algorithm. In this technique the graph representation of data is based upon two approaches- the object similarity and number of shared neighbours between objects. Besides this a Markov chain model is built upon this graph, which assigns an outlier score to each object. Agrwal[21] has suggested a local subspace based outlier detection which uses different subspace for different objects. This approach basically adopts local density based outlier detection by defining a Local Subspace based Outlier Factor (LSOF) in high-dimensional datasets. A. Ghoting (et. al)[23] proposed an outlier detection algorithm, LOADED, for outlier detection in evolving datasets containing both continuous and categorical attributes. LOADED is a tuneable algorithm, wherein one can trade off computation for accuracy so that domain-specific response times are achieved. S. Wu (et. al)[24] incorporated the concept of entropy to propose an information theoretic outlier detection technique for large-scale categorical data. This strategy, first, adopts a deviation-based strategy, avoids the use of statistical tests and proximity-based measures to identify exceptional objects. Secondly, combine entropy and total correlation with attribute weighting to define the concept of weighted holo-entropy, where the entropy measures the global disorder of a data set and the total correlation measures the attribute relationship.

2.1. Discussion and Motivation

From the inception of research on outlier detection in data mining, researchers have focussed on most trivial distance based approach to most recent ranking driven approach[20] of outlier detection. In course of time, several contextual modifications are made on density-based, graph-based and statistical outlier detection approaches, but none is able to provide a very acceptable solution, with a high accuracy, to the outlier detection problem. To summarize, based on our survey we observe the following.

- Although distance based approach is a trivial criteria for outlier detection, but it alone is not suitable for the datasets having clusters of different distribution.
- In the distance based outlier detection[15], the main overhead is the selection of the user-defined fraction of data those are further away than another user-defined threshold distance.
- The Statistical approaches require either construction of

a probabilistic data model based on empirical data, which is rather a complicated computational task, or require a priori knowledge of the distribution laws. Even if the model is parameterized, complex computational procedures for finding these parameters are needed. Moreover, it is not guaranteed that the data being examined match the assumed distribution law if there is no estimate of the density distribution based on the empirical data.

- Density based approach of outlier detection considers neighbourhood density of points to declare outlier or non-outlier. This approach is able to provide better detection results if selection of the required input parameter ε is done accurately.
- The performance of the existing outlier detection algorithms are dataset dependent. Therefore, development of a robust, sensitive outlier detection technique which is free from the limitations offered by the aforesaid algorithms is of utmost importance.

3. Background of the Work

In this section, we will discuss the background concepts which provide the basis of our work. The proposed outlier detection technique is a combination of both distance and density based outlier detection approach along with the concept of entropy as a measure for outlier detection.

3.1. Outlier

Outliers are those observations in the data that do not conform to the inherent patterns of the data. There are several definitions given for outlier from different view point. An example of outliers in two dimensional dataset is illustrated in Figure 1. Outliers may be induced due to a variety of reasons such as malicious activity (e.g., credit card fraud, cyber attacks, novelty detection, and breakdown of a system), but all these reasons have a common characteristic that they are interesting to the analyst. The interestingness or real life relevance of outliers is a key feature of outlier detection[19]. Outlier detection is related to, but distinct from noise removal or noise accommodation that deals with unwanted noise in the data. Noise does not have any real life significance and acts as hindrance to data analysis.

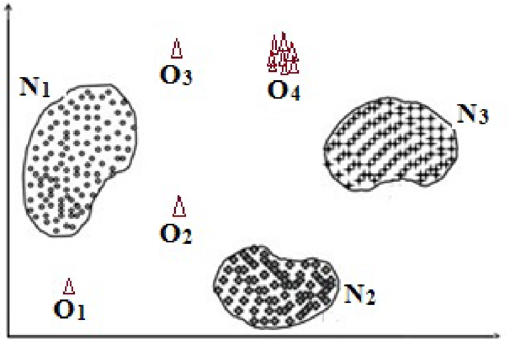

Figure 1. Outliers in two dimensional dataset: N_1, N_2, and N_3 are the three normal regions. Points that are sufficiently far away from the normal region (e.g., points O_1, O_2, O_3 and points in O_4 regions) are outliers

3.2. Distance-based Outlier

Distance-based method was originally proposed by Knorr and Ng[15]. It states that -"*An object O in a dataset T is a DB(p, D)-outlier if at least fraction p of the objects in T lies greater than distance D from O*". This notion is further extended based on the distance of a point from its k-th nearest neighbour. Alternatively, the outlier factor of each data point is computed as the sum of its k-th nearest neighbours. Here the distance can be proximity given by any of the dissimilarity measure *Euclidian distance*, *L_p norm*, *Cosine distance* etc.

3.3. Density-based Outlier

Density-based approach was proposed by Breuning et al.[4]. It relies on the local outlier factor (LOF) of each point, which depends on the local density of its neighbourhood. In our work we considered local neighbourhood density in terms of number of points lying in the ε-neighbourhood of the object. In this view point, an outlier is the point lying so sparsely that there are not more than a threshold *MinPts* number of other points lying in the ε-neighbourhood of that point.

3.4. Entropy

In information theory, entropy is core concept that measures uncertainty about a stochastic event and it means that entropy describes the distribution of an event[13]. Entropy is a measure of disorder or more precisely unpredictability in a system. In entropy-based clustering, an object is added to that cluster such that upon addition the increase in intra-cluster entropy is minimum among all other clusters. Since outlier is the observation that deviates from the inherent pattern of the data, so upon addition of such point to any cluster in the dataset, the increase in entropy is much higher than a non-outlier point. This notion is an important criterion for entropy based outlier detection. Shannon denoted the entropy H of a discrete random variable X with possible values $\{x_1,x_2....., x_n\}$ and probability mass function $p(X)$[8] as,

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_b p(x_i) \tag{1}$$

where b is the base of the logarithm used. Previously, entropy has been a metric difficult to evaluate without imposing unrealistic assumptions about the data distributions[13]. Renyi proposed an entropy measure that lends itself to nonparametric estimation directly from data[13]. The mathematical formula for Renyi's entropy is briefly described in *section 5.4*.

4. Locality Sensitive Hashing (LSH)-based Outlier Detection

This is a distance based approach in conjunction with a ranking scheme based on the concept of Locality Sensitive

Hashing (LSH)[5]. The basic idea for LSH is to convert the data into manageable fingerprints and hash them so that similar data points are mapped to the same buckets with high probability.

Definition: A family H is called (R, c, P_1, P_2)-sensitive if for any two points $p, q \in \Re$ [5]

$$\text{If } \|p-q\| \leq R \quad then \Pr_H[h(p)=h(q)] \geq P_1 \tag{2}$$

$$\text{If } \|p-q\| \geq cR \; then \Pr_H[h(p)=h(q)] \leq P_2 \tag{3}$$

The first condition guarantees that similar points are hashed to the same bucket with high probability whereas the second condition says that distant points are hashed to the same bucket with small probability. A family will be useful only when $P_1>P_2$. In order to improve the efficiency of the outlier detection process some pruning techniques are used viz. PPSO, ANNS[20]. The whole framework of LSOD can be divided into a number of modules. The initial step is effectively a pre-processing step in which the dataset is divided into a number of clusters. So, the exact clustering technique employed is independent of the outlier detection framework. Further steps are briefly described below.

4.1. Outlier Likelihood Ranking

The points in the database are first ranked based on their likelihood to be an outlier. The resulting rank-ordered list is then processed in the detection phase where the actual outliers are found. The intuition behind this outlier ranking order is-*lower the rank, higher is the likelihood to be an outlier*. The outlier likelihood rank of any object is given by a ranking function called LSH function $h(v)$ that leverages p-stable distribution[20][5].

$$h(v) = \left\lfloor \frac{a_i.v + b_i}{w} \right\rfloor \tag{4}$$

where w is a parameter of the hashing procedure that denotes the size of the windows onto which the database points are projected. It is generally recommended that w=4[20]. a_i is a d-dimensional vector and the value of each dimension is drawn from the standard normal distribution. b_i denotes a random bias whose value is drawn from the uniform distribution *Unif(0,w)*. The probability $p_q(d)$ of a point p that is at a distance d from another point q and is hashed to the same bucket is given by,

$$p_q(d) = \Pr[h_i(q) = h_i(p)] = \int_0^w \frac{1}{d} f_p(\frac{t}{d})(1-\frac{t}{w})dt \tag{5}$$

Where

$$f_p(t) = \int_0^t \sqrt{2/\pi} \exp(\frac{-x^2}{2})dx \tag{6}$$

In Equation 6, $f_p()$ is a strictly increasing function. For a fixed parameter w in Equation 5, $p_q(d)$ decreases monotonically with d. In other words, the collision probability between points p and q decreases as the distance $\|p-q\|$ between them increases. The performance of the locality sensitive hashing (i.e. the hash family H) depends on the parameter R, which is an estimate of distance between a normal point and one of its neighbours. There exists several research efforts focusing on addressing issues related to LSH parameter tuning[2][7]. The LSH-based outlier detection[20] relies on ranking for efficiency, not correctness. Therefore, a slightly less accurate ranking will not significantly impact performance of outlier detection. However, an efficient approach for estimation of R is employed in[20] based on the already generated clusters. First some pairs of points are sampled, where each pair of points are in the same cluster, and then calculate the distances between these pairs. Finally, set the median of these distances as the estimated value of R.

4.2. Ranking Methodology

For a given point q, let N_q denotes the number of points that hash to the same bucket as q. We define rank(q) as follows[20]

$$\text{rank}(q)=\text{E}[\text{N}_\text{q}] \tag{7}$$

where $E[N_q]$ is the expected number of points in the database that hash to the same bucket as q. We can formally define $E[N_q]$ as follows[20]:

$$E(N_q) = \sum_{u \in D} 1.p_{collision}(u,q) \text{ that is,}$$

$$E(N_q) = \sum_{u \in D} p_{collision}(u,q) \tag{8}$$

4.3. Outlier Detection

After the ranking is over, the objects are processed in an increasing order or rank. This ordered ranking scheme has the advantage of processing most probable outlier candidates first. Again, based on the weakest outlier score and user defined parameter k, first L number of outliers is returned as output.

Apart from the LSH-based outlier detection technique[20], we have compared our proposed outlier detection technique, RODHA, with three other outlier detection techniques viz. LOF[4], ORCA[3] and *OutRank-b*[16]. The Table 1 shows a general comparison of these four existing outlier detection techniques.

5. RODHA: The Proposed Outlier Detection Technique

RODHA (Robust Outlier Detection using Hybrid Approach) is designed using a combination of both distance and density based outlier detection approach in conjunction with entropy measure from information theory. The basic framework of the RODHA is shown in the figure 2. It requires clustering of the data as a pre-processing step. Then the distance based approach defines an object to be an outlier when its minimum distance from all the cluster profiles is greater than the maximum intra-cluster distance of all the clusters in the data. The density-based approach to outlier detection relies on the computation of the local neighborhood density of a point. Implicit to this approach is the notion of distance but an additional criterion is that of neighborhood and the determination of number of points lying within a neighborhood of interest. Finally, the notion of entropy based outlier is that a candidate outlier sample when added to its nearest cluster would increase the intra-cluster entropy by an amount much higher than a non-outlier sample when added to the same cluster.

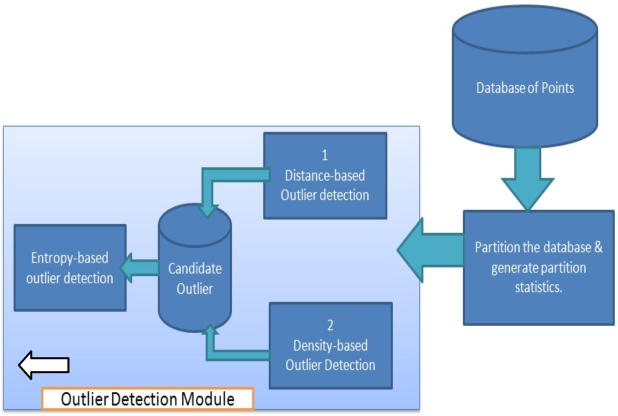

Figure 2. Basic Framework of RODHA

Let us consider D be the database of points from $\Re^d$ where d is dimension of the points in the database. As a pre-processing step, the dataset D is divided into two parts, a large training set $D_{trainset}$ and a smaller test set $D_{testset}$. The overall framework of the proposed technique consists of four phases as follows.

```
ALGORITHM 1: K-Means-with-Farthest-seeds Algorithm
Input: $k$: the number of clusters, $D_{trainset}$: a data set containing $N$ objects.
Output: A set of $k$ clusters
Procedure:
Step-1: Call Farthest-$k$-Object($D_{trainset}$, $k$);
Step-2: Repeat
Step-3: (re)assign each object to the cluster to which the object is the most similar,
based on the mean value of the objects in the cluster;
Step-4: Update the cluster means, i.e., calculate the mean value of the objects for
each cluster;
Step-5: Until no change;
```

5.1. Clustering the Training Set

The given training set $D_{trainset}$ is clustered to produce k number of clusters $C_1, C_2,......C_k$ and accordingly the objects are labeled. Although clustering is a pre-requisite to the outlier detection framework, the final outlier detection is independent of the exact choice of the clustering method. We have used *k-means* clustering algorithm 1[11]. One can employ other popular clustering methods like *k-medoid*, *DBSCAN, fuzzy c-means* etc clustering methods. The performance of *k-means* clustering depends heavily on the selection of initial cluster centroid. So, we have employed a routine for selecting the farthest k objects as the initial cluster centroid.

5.1.1. Initial Centroid Selection

The farthest k objects from the training set, $D_{trainset}$ are selected as the initial centroid in the *k-means* clustering algorithm. The first centroid is selected randomly from the training set $D_{trainset}$. The point farthest from the first selected point is selected as the second initial cluster centroid. Then, the next centroid is the point in the training set, $D_{trainset}$, for which the sum of its distance from all the already selected centroid is maximum. This process continues until all the user defined number of centroids are selected.

For datasets having very high dimensions, the farthest-k-objects selection using the above distance based approach suffers from curse of dimensionality. For those datasets, it is preferable to employ spatial index structures like R-tree or its family members, because it can reduce-

(a) the cost of neighbourhood computation as average case complexity of searching in R-tree is $O(log_m n)$, where n is the total number of nodes in the R-tree and m is the number of entries in a node, and

(b) the cost of finding the farthest point significantly.

```
ALGORITHM 2: Farthest-k-Object
Input: $D_{trainset}$: a data set containing $N$ objects. $k$: the number of clusters,
Output: Set InitCent of k farthest objects
Procedure:
Step-1: Initialize $InitCent=\phi$;
Step-2: Randomly choose any of the objects from $D_{trainset}$ as first point $C_{init1}$;
Step-3: $InitCent=InitCent \cup \{C_{init1}\}$
Step-4: $C_{init2}$=Farthest object $O \in D_{trainset}$ from $C_{init1}$
Step-5: $InitCent=InitCent \cup \{C_{init2}\}$
Step-6: If k=2, return InitCent; exit; Else
Step-7: $i$=3;
Step-8: Repeat
Step-9: $C_{initi}$= Object $O \in D_{trainset}$, such that sum of its distance from all the points in InitCent is maximum;
Step-10: $InitCent=InitCent \cup \{C_{initi}\}$
Step-11: i=i+1;
Step-12: Until $i=k$.
```

5.2. Distance based Outlier Detection

Let us consider the dataset D has the spatial distribution as shown in the figure 3, i.e. the clusters $C_1, C_2,......C_k$ in the data set are of convex nature being well-separated from one another. Let $C_{C1}, C_{C2},......C_{Ck}$ be the respective centroids of the k clusters in the dataset D. Find the maximum distances $d_{max1}, d_{max2},......d_{maxk}$ of the centroids to the cluster objects. Then identify the threshold distance d_{thres} as the maximum of the values $d_{max1}, d_{max2},......d_{maxk}$. For a test object O, find d_{min}, the minimum of the distances $d_{OC1}, d_{OC2},......d_{OCk}$ of the object O to the centroid $C_{C1}, C_{C2},......C_{Ck}$. If d_{min} is greater than d_{thres}, then the test object O is identified as an outlier, otherwise it is a normal object.

Table 1. Existing Outlier Detection Techniques: A General Comparison

	LSH-OD[20]	LOF[4]	ORCA[3]	OutRank-b[16]
Input Parameter	K, L	k, MinPts	K, N	∈
Approach	Distance and Ranking based	Density based	Distance based	Stochastic Graph Based
Data Set Applied	KDD99,Covertype etc.	Hockey, Soccer etc.	Corel, Covertype etc.	UCI KDD achive
Complexity	$\theta(\lvert D\rvert.\dim.\lvert H\rvert)$	O(n^2)	O(n^2)	-

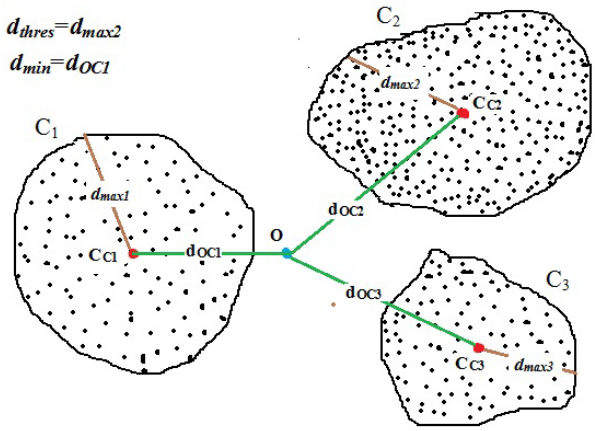

Figure 3. Illustrating distance-based outlier detection approach. C_1, C_2, C_3 are three normal clusters in the data, Outlier point O's minimum distance from all three clusters d_{min}=d_{OC1} is greater than maximum intra-cluster distance d_{thres}=d_{max2}

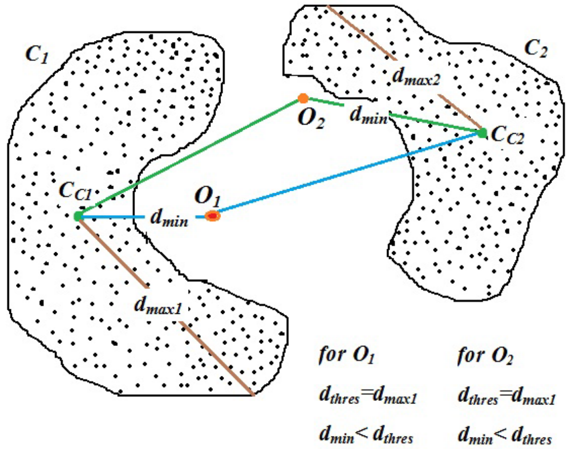

Figure 4. Illustrating distance-based outlier detection approach fails in case of clusters being concave in nature. C_1, C_2 are two normal clusters in the data, Outlier point O_1's minimum distance from the two clusters d_{min}=d_{OC1} is not greater than maximum intra-cluster distance d_{thres}=d_{max1}

This distance-based approach can detect outliers where the dataset is of convex in nature. But the approach fails for datasets of concave nature (as shown in figure 4) or where the outlier objects are lying marginally away from the boundary of the clusters. In figure 4, the two clusters C_1, and C_2 are of concave nature. The object O_1 is supposed to be an outlier. By distance based approach we find d_{thres}=d_{max1} and d_{min}=d_{OC1}. But the condition for being an outlier i.e. d_{min}>d_{thres} is not satisfied by O_1. So, the distance approach fails to detect O_1 as outlier. Same situation arises for object O_2 lying close to the boundary of the cluster C_2. To handle such situation our proposed technique employs density based approach to detect outlier.

5.3. Density based Outlier Detection

The density based approach requires selection of a parameter value ε which is the distance to check the availability of any training samples within the ε-neighbourhood of the test sample *O*.

Definition (ε -neighbourhood): For an object *O*, ε-neighbourhood finds all the samples within a distance of ε from the object *O*.

For a test object *O*, first find all the points in the ε-neighbourhood of *O*. Followed by this, check whether these neighbouring points are already labelled. If there are no points in the ε-neighbourhood of *O* that are also labelled, i.e. the object *O* lying much away from the boundary of any cluster in the data set, then the object *O* is a candidate to be an outlier. One such situation is shown in the figure 5.

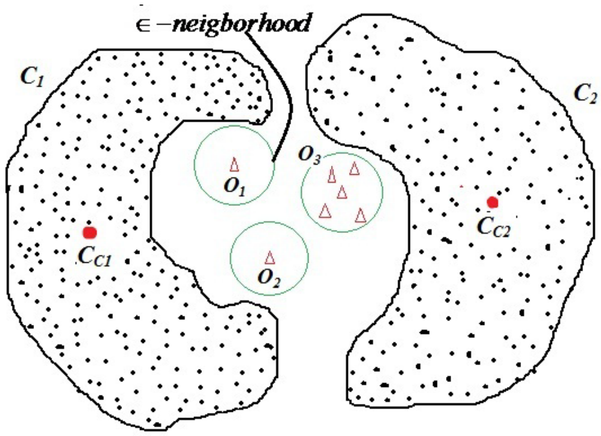

Figure 5. illustrating density-based outlier detection approach. C_1, C_2 are two normal clusters in the data, the green circle around O_1 and O_2 shows their ε-neighbourhood within which no cluster-labelled points lying. O_3 depicts a too small group of objects to form a cluster and within ε-neighbourhood lies no cluster labeled points. So, O_1, O_2 and groups of O_3 are all outliers

Here the objects O_1 and O_2 are the candidate outliers. Within the ε-neighbourhood of O_1 and O_2 there are some points, but none are labelled i.e. none are the points included within any of the clusters in the data set. One very important aspect of this approach to be noted is the selection of the ε-value. The performance of this approach is very sensitive to the selection of the ε-value. If we take a larger ε-value, then some of the candidate outlier points might not be detected. Again the ε-value for an object should not be too small such that it finds no labelled points within the distance ε-neighbourhood and erroneously declares it to be outlier. So, the value of the parameter should be selected experimentally. In the section 6, we have provided a heuristic method of selecting ε-value for some UCI Machine learning datasets[6] on which we have applied our technique.

5.4. Entropy as a Parameter for Outlier Detection

In information theory, entropy is core concept that measures uncertainty about a stochastic event and it means that entropy describes the distribution of an event[13]. Entropy is a measure of disorder or more precisely unpredictability in a system. In the field of data mining, entropy has been used in clustering applications[13] where an object is included to that cluster where its inclusion increases the intra-cluster entropy or disorder by minimum. The notion of entropy in this perspective provides important criteria for outlier detection. By the definition, an outlier is an instance that is much different from the inherent pattern of the data. Such an object instance when added even to its nearest alike cluster would increase the amount of intra-cluster entropy much higher than a non-outlier object when added to the same cluster.

There are several mathematical formulations exist to measure entropy of a system. One of the very popular methods of entropy is Shannon entropy. We have employed an effective entropy measure known as "Renyi's Entropy"[13][17]. It is a generalized form of Shannon entropy developed by Alfred Renyi.

Definition (Renyi's Entropy): Renyi's entropy for a stochastic variable X with probability density function (pdf) f_x is given by

$$H_R(X) = \frac{1}{1-\alpha}\log\int f_x^{\alpha}dx \quad \alpha > 0, \alpha \neq 1 \tag{9}$$

Specifically for $\alpha = 2$ we obtain,

$$H_R(X) = -\log\int f_x^2 dx \tag{10}$$

This is called Renyi's entropy. The expression can easily be estimated directly from data by the use of Parzen window estimation, with a multidimensional Gaussian window function. Assume that cluster C_k consists of the set of discrete data points x_i , $i=1,2,....N_k$. Now, the probability density estimate based on the data points of C_k, is given by[13][26]

$$\hat{f}_k = \frac{1}{N_k}\sum G(x - x_i, \sigma^2 I) \tag{11}$$

where N_k is the number of data points in C_k, and we have used symmetric Gaussian kernel of covariance matrix $\sum = \sigma^2$ I. By substituting equation 11 into equation 10 and utilizing the properties of the Gaussian kernel, we obtain an estimate of the entropy of C_k as

$$H(C_k) = -\log V(C_k) \quad \text{where} \tag{12}$$

$$V(C_k) = \frac{1}{N_k^2}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} G(x_i - x_j, 2\sigma^2 I) \tag{13}$$

Since the entropy is calculated based on points assigned to the same cluster, we refer to 12 as the within-cluster entropy. We have used entropy based clustering as a support to detect outliers and have found that there is a very high increase in the within-cluster entropy for the points that are detected as outliers.

5.5. Complexity Analysis

The framework of the proposed outlier detection approach uses clustering as a pre-processing step. We have used *k-Means-with-Farthest-seed* for clustering which takes *O(N*k*I)* time, where *N* is the total number of objects, *k*, the number of clusters and *I*, the number of iterations. The *Farthest-k-Object* routine for initial cluster centroid selection takes an *O(N)* time. The within-cluster entropy can be calculated in *O(N_c)* operation for a cluster with N_c number of points. Followed by this, in Step-4 of the algorithm 3, for a N_c number of intra-cluster points the distance of centroid from all intra-cluster points take *O(N_c*dim)*, where *dim* is the dimension of the dataset, and searching for maximum of maximum intra-cluster distances take linear time complexity

In Step-6, for the test dataset *D_{test}[m*dim]*, the distance from all test objects to centroids take *O(m* k*dim)* time and the searching for minimum among all the distance take a linear *O*(*k*) time. While searching for ε-neighbourhood of O_i, the algorithm considers objects from both training and test datasets and calculates the distances. So, the complexity in this step is *O*((*m*+*N*)**dim*). Finally, in Step-10 to Step-12, adding each candidate point to nearest cluster and then caculating the updated entropy takes *O*(N_c+1). Thus the overall complexity of the proposed outlier detection technique is *O*(*N***k***I* + *m***k***dim* + (*m*+*N*)**dim*) + *O*(N_c). For a large dataset with small number of clusters, we can ignore *k*, then the overall complexity becomes *O*(*N***I*+*N***dim*).

6. Empirical Evaluation

6.1. Environment Used

The proposed RODHA algorithm is implemented in a computer system with processor Intel(R) Core(TM) 2 Quad CPU Q6600 @ 2.4 GHz and RAM of 2 GB in a 32-bit Windows 7 operating system. The algorithms are programmed in programming language C (Borland C++ version 4.5) and Matlab (version 7.6.0 R2008a).

6.2. Datasets Used

Several synthetic and real life datasets are used for testing the performance of the proposed RODHA and the LSH-based outlier detection algorithm. The real-life datasets are downloaded from UCI Machine Learning Repository website. In the Table 2, we have provided the basic information about the data sets considered for our experiments. For the data sets having missing values, prior to implementing the outlier detection algorithms, we estimated the missing values using two very popular estimation techniques- KNN Imputation[18] and LLS Imputation[14].

```
ALGORITHM 3: RODHA Algorithm
Input: $D[N*dim]$:Database of $N$ data objects of $dim$-dimensions; $k$: number of clusters; $D_{test}[m*dim]$: Test dataset; $\varepsilon$: the $\varepsilon$-neighbourhood; $\tau$: Threshold for intra-cluster entropy difference;
Output: Return the outliers in $D_{test}[m*dim]$.
Procedure:
Step-1: Initialize set $CandidateOutlier=\phi$;
Step-2: $C$=k-Means-with-Farthest-seed($D[N*dim]$, $k$, $Iter$) such that $C=\{C_1, C_2,......C_k\}$ and $C_{centroid}=\{C_{C1}, C_{C2},......C_{Ck}\}$ respectively. Label the objects accordingly.
Step-3: Find within-cluster entropy $\{E_1, E_2,......E_k\}$ for all $C_1, C_2,......C_k$ respectively.
Step-4: Find maximum value of the maximum distances $d_{max1}, d_{max2},......d_{maxk}$ of centroid to the cluster objects for each cluster and identify it as threshold distance $d_{thres}$ i.e.
        $d_{thres}= max(d_{max1}, d_{max2},......d_{maxk})$
Step-5: Repeat
    Step-6: for a test object $O_i$ in $D_{test}[m*dim]$, Calculate $d_{iC1}, d_{iC2},......d_{iCk}$ the distance of $O_i$ from centroid $C_1, C_2,......C_k$. Assign $d_{min}=min(d_{iC1}, d_{iC2},......d_{iCk})$. If $d_{min}>d_{thres}$, add $O_i$ to the set $CandidateOutlier$. Else
    Step-7: For the test object $O_i$ in $D_{test}[m*dim]$ find all points in its $\varepsilon$-neighbourhood, i.e $\varepsilon\text{-}nbr(O_i)=\{ p_1, p_2,.......p_n \}$. If no point in $\varepsilon\text{-}nbr(O_i)$ is labelled by clusters $C_1, C_2,......C_k$, then add $O_i$ to the set $CandidateOutlier$.
Step-8: Until all the test objects in $D_{test}[m*dim]$ are tested.
Step-9: Repeat
    Step-10: for an object $O_i \in CandidateOutlier$, add $O_i$ to the nearest cluster $C_j$ (j=1or 2 or....or $k$). Find the updated intra-cluster entropy $E_j'$ for $C_j$.
    Step-11: Find the difference, $diffEntropy=E_j'-E_j$. If $diffEntropy \geq \tau$, Report $O_i$ an outlier.
Step-12: Until all the objects in $CandidateOutlier$ are tested.
```

Table 2. Basic information of the datasets used

Datasets	No. of Instances	No. of Attributes	Attribute Type	No. of Classes	Missing Values
Synthetic-1	160	4	Real	3	No
Abalone	4177	8	Categorical, Real, Integer	29	No
Iris	150	4	Real	3	No
Wine	178	13	Integer, Real	3	No
Statlog Heart	270	13	Categorical, Real	2	No
eColi	336	8	Real	8	No
Yeast	1484	8	Real	10	No
Glass Identification	214	10	Real	2	Yes
Pima	768	8	Integer, Real	2	Yes
Housing	506	14	Categorical, Real, Integer	5	No
Breast Cancer	569	32	Real	2	Yes
Vehicle	946	18	Integer	4	No
Sonar	208	60	Real	2	No
Zoo	101	17	Categorical, Integer	7	No
Ionosphere	351	34	Real, Integer	3	No
Habermans Survival	306	3	Integer	2	No
Hayes Roth	160	4	Categorical, Real	3	No
Liver Disorder	345	7	Categorical, Real, Integer	2	No
Teaching Asst. Evaluation	151	5	Categorical, Integer	3	No
Cloud	1024	10	Real	2	No
Hill Valley	606	101	Real	2	No
Libras Movement	360	90	Real	24	No
Concrete Slump Test	103	10	Real	2	No
Vertebral Column	310	6	Real	4	No

Table 3. Performance Evaluation of RODHA with other Competing Algorithms in term of Detection Rate (DR)

Datasets	LOF[4]	ORCA[3]	OutRank-b[16]	LSH-OD[20]		RODHA		
	DR	DR	DR	k	DR	ε	τ	DR
Synthetic-1	0.7500	0.8500		17	1.0000	0.3	0.0865	0.9500
Abalone	0.8902	0.8691		30	0.9928	0.7	0.1741	0.9974
Iris	0.8911	0.8633	-	13	0.8733	0.52	0.2214	0.9467
Wine	0.9233	0.9122	-	18	0.9045	2.4	0.4370	0.9607
Statlog Heart	0.9108	0.8969	-	15	0.9370	4.23	0.6015	0.9593
Glass Identity	0.8813	0.8388	-	13	0.9299	1.7	0.2144	0.9500
Breast Cancer	0.8643	0.8109	-	40	0.9666	3.21	0.7365	0.9807
Sonar	0.8800	0.8477	-	17	0.9038	1.2	0.3282	0.9231
Pima	0.9333	0.9041	-	15	0.9701	0.64	0.1674	0.9779
Zoo	0.8235	0.8823	-	17	0.8317	3.55	0.5144	0.9406
Vehicle	0.3095	0.2919	1.0000	20	0.9789	0.51	0.2029	0.9400
eColi	-	-	1.0000	15	0.9524	0.95	0.2921	0.9613
Yeast	-	-	0.6428	20	0.9865	1.1	0.1245	0.9798
Housing	-	-	-	15	0.9585	1.72	0.5467	0.9768
Ionosphere	-	-	-	30	0.8433	8.65	0.4509	0.9487
Haberman's Survival	-	-	-	17	0.9314	0.21	0.1800	0.9510
Hayes Roth	-	-	-	15	0.8788	1.94	0.1500	0.9091
Liver Disorder	-	-	-	19	0.9536	1.23	0.1645	0.9681
Teaching Asst. Eval.	-	-	-	14	0.8675	1.24	0.4712	0.9404
Cloud	-	-	-	30	0.9180	0.65	0.1180	0.9297
Hill Valley	-	-	-	17	0.9604	1.67	1.9200	0.9571
Libras Movement	-	-		18	0.9083	4.83	1.9980	0.9250
Vertebral Column	-	-		24	0.8901	0.68	0.1893	0.9355
Concrete Slump Test	-	-		10	0.8447	1.98	0.4322	0.8738

6.2.1. KNN-Impute

The k-Nearest Neighbour based method selects samples similar to the sample of interest to impute missing values. If we consider sample O that has one missing value in first attribute, this method would find k other objects, which have a value present in attribute 1, with attribute values most similar to O in attributes N-2 (where N is the total number of attributes). A weighted average of values in attribute 1 from the k closest samples is then used as an estimate for the missing value in sample O. In the weighted average, the contribution of each sample is weighted by similarity of its attributes to that of sample O. For similarity metric we can employ the proximity measures like *Euclidian Distance, Pearson Correlation Coefficient* etc.

6.2.2. LLS-Impute

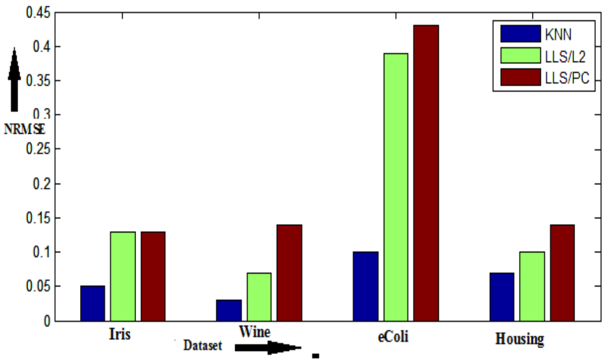

Figure 6. Performance of Imputation methods in terms of *NRMSE*

The Local Least Squares Imputation method (LLS-Impute) represents a target sample that has missing values as a linear combination of similar samples. The similar samples are chosen by k-nearest neighbours that have large absolute values of Pearson correlation coefficients. The imputation can be performed, regardless of how the k samples are selected. Thus, both correlation coefficient and and L_p norm can be used for k-nearest neighbour selection.

The performance in terms of *Normalized Root Mean Square Error* (*NRMSE*) of KNN-Impute, LLS-Impute/L2 and LLS-Impute/PC over four datasets each having missing values in 20% objects are shown in the figure 6.

Definition (*NRMSE*): The Normalized Root Mean Square Error (*NRMSE*) is defined as:

$$NRMSE = \sqrt{mean[(y_{guess} - y_{ans})^2]} \Big/ std[y_{ans}] \quad (14)$$

where y_{guess} and y_{ans} are vectors whose elements are the estimated values and the known answer values respectively, for all missing entries. The mean and the standard deviation are calculated over missing entries in the entire matrix.

6.3. Experimental Results

As mentioned earlier, we have implemented the proposed outlier detection algorithm (RODHA) and the LSH-based outlier detection[20] on the datasets tabulated in Table 2 from UCI Machine Learning data set archives. The RODHA algorithm is compared with four other outlier detection algorithms-LSH-OD[20], ORCA[3], LOF[4], and *OutRank-b*[16]. These techniques are compared based on the detection rates of outliers. The detection rate (DR) is calculated based on the ROC analysis proposed in[9].We have downloaded the executable versions of LOF[1] and ORCA[3]. Results of LOF and ORCA are also reported for these datasets in the columns 2 and 3 respectively. The detection rate values for *OutRank-b* are taken from the paper[16]. The comparison of these five different techniques over 14 different synthetic and benchmark datasets is shown in the Table 3.

6.4. Discussion

The effectiveness of the proposed technique depends upon the value of the user defined parameter ε and that of LSH-based technique depends upon user defined parameter k. So, in the last four columns of the Table 3, we have reported the detection rates of LSH-OD and the proposed technique RODHA along with the value of respective user defined thresholds (k and ε). Out of the 24 datasets (1 synthetic and 23 benchmark UCI datasets) except for (Vehicle and Hill Valley) where detection rate is slightly less, the proposed outlier detection algorithm shows excellent performance over all other datasets.

6.5. Entropy Measure as an Outlier Detector

In our proposed technique of outlier detection we have used entropy from Information theory as a support to detect outlier. When one candidate outlier sample is added to its nearest cluster, it increases the within cluster entropy significantly than a non-outlier sample. So, we have used this significant increase of entropy as a weightage to declare the object an outlier. Figure 8 shows the effectiveness of entropy for outlier detection for a synthetic data set of figure 7. The bar diagram (figure 8) shows the change (i.e. increase) in intra-cluster entropy for the objects in the test set. The longer bars are the increase in entropy for outlier points that are much higher than the rest non-outlier points in the test dataset.

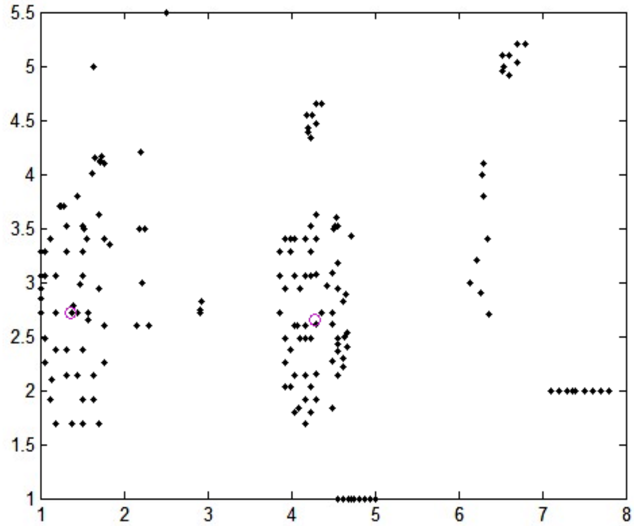

Figure 7. A 2-Dimensional synthetic data set containing two clusters and outliers

6.6. Sensitivity of the proposed Algorithm

In order to test how sensitive the proposed algorithm towards detecting outliers, we have made a synthetic dataset as shown in the figure 9.The points in the synthetic dataset is distributed in six different ways. Here N_1, N_2 are two normal clusters, O_1 is distinct outlier, O_2 is distinct inliers, O_3 is equidistant outlier, O_4 is border inlier, O_5 chain outlier, O_6 is compact group of objects too small in numbers to form a cluster and O_7 is outlier of "stay together" nature. To test the effectiveness of the algorithm, we label these as candidate outliers and run the outlier detection algorithm. The outliers returned by the technique is compared with the candidate outliers and we find the accuracy of the detection equal to 0.98 i.e. the algorithm is sensitive to such outliers with an accuracy level of 98%.

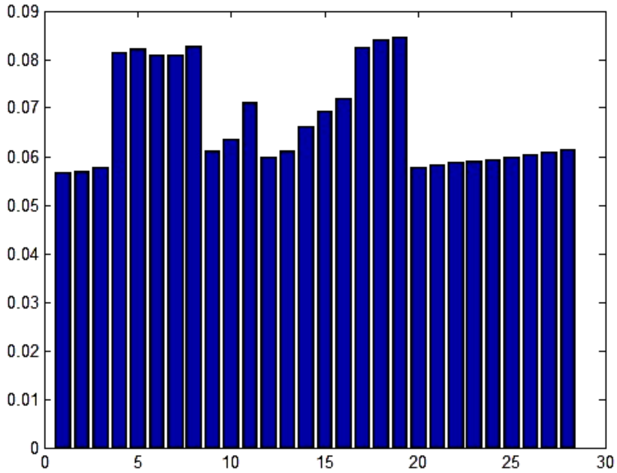

Figure 8. Illustrating the increase in entropy for addition of test dataset objects in the nearest cluster. The bars with longer height are the entropy increase for outlier points

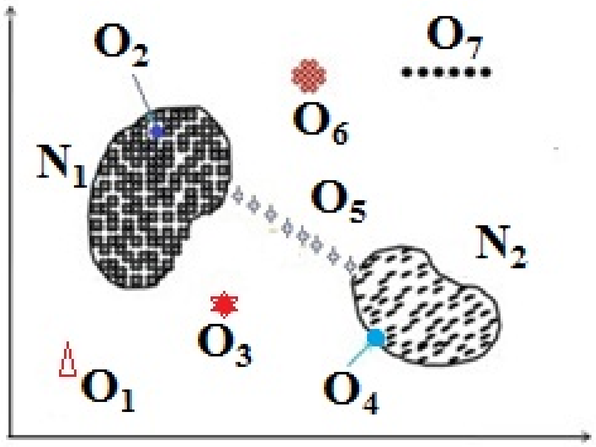

Figure 9. Illustration of six different cases: N_1 and N_2 are two normal clusters, O_1 is the distinct outlier, O_2, the distinct inlier, O_3, the equidistance outlier, O_4, the border inlier, O_5, a chain of outliers, O_6 is another set of outlier objects with higher compactness among the objects and O_7 is an outlier case of "stay together"

6.7. Selection of ε threshold in the proposed RODHA Algorithm

The effectiveness of the proposed outlier detection algorithm depends on the choice of the value for the threshold ε , which is a neighbourhood distance. The threshold value varies from data set to data set and this variation is very high since the datasets varies in number, type and range of attribute values. So, in order to provide a relatively common range of selecting the threshold value for different data sets, we normalize the data set attribute value within a range of 1.0 to 5.0 as a pre-processing step. While normalizing data within this range we discarded the attributes that take binary values or some fixed number of constant values. In the following figure 10 we have provided a heuristic method of selecting the ε-value where the detection rate is plotted against the ε-value for different datasets and the accuracy of detection is maximum for the threshold (ε) value within the range 0.4 to 1.5.

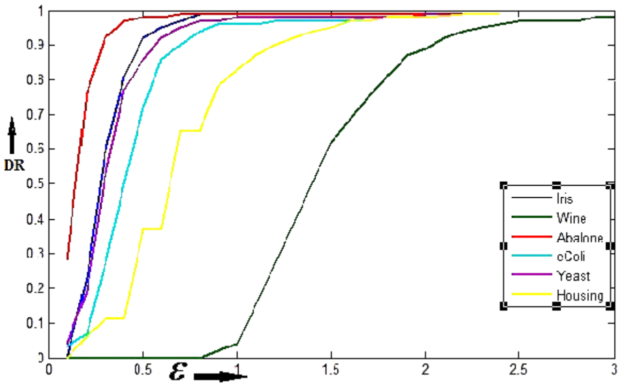

Figure 10. Illustrating the accuracy of outlier detection for different values of ε in RODHA. It is useful for identification of threshold ε range values, where it performs well

6.8. Selection of threshold τ in the proposed RODHA Algorithm

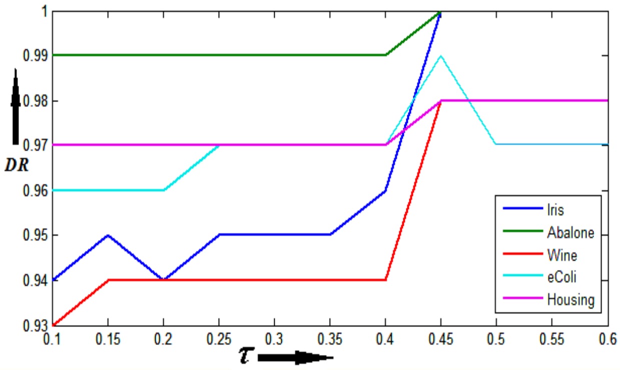

Figure 11. Illustrating the variation of outlier detection rate against the user-defined threshold τ in RODHA

As mentioned earlier, proposed outlier detection technique, employs entropy as a weighage to declare an object to be an outlier. Here, for a candidate outlier detected in distance and density based outlier detection phase of the algorithm, are made to pass through another entropy based outlier detection phase, where the the object is added to its nearest cluster and the resulting increase in the intra-cluster entropy is compared to an entropy difference threshold τ. The proposed outlier detection technique depends on both the user-defined thresholds- ε and τ. In the figure 11, shows the variation of detection rate against the values of user defined threshold τ for some datasets. The detection rate of the algorithm stays almost steady between 0.93 to 0.97 for value of τ between 0.2 to 3.5, beyond which although it shows higher detection rate, but τ-value becomes so large that practically no outliers get detected.

6.9. Selection of threshold k in LSH-based outlier Detection Technique

The performance of the LSH-based outlier detection scheme largely depends on the selection of k. Again, since the datasets are different in terms of number of objects, attribute value and range of attribute values, so the value of k also varies accordingly. However, here we have employed a heuristic method to find a most probable range of k leading to better result. In figure 12, we have shown the variation of accuracy (detection rate) of the algorithm with different value of k. We see that for the standard datasets namely Iris, Breast Cancer Wisconsin, Statlog Heart, e-Coli, Yeast, Housing and Wine the accuracy remains almost steady between 96% to 99% for value of k from 10 to 25. Beyond k value 25, the performance decreases significantly. Again although, for value of k between1 to 10 the graph shows the accuracy level of around 98%, but in that range the number of outlier detection is quite low. So, we consider that range of the value of k for which the accuracy level is relatively better and also the algorithm detects the probable outliers. In our experiments, better result of the outlier detection algorithm is found with k value in the range of 10 to 25.

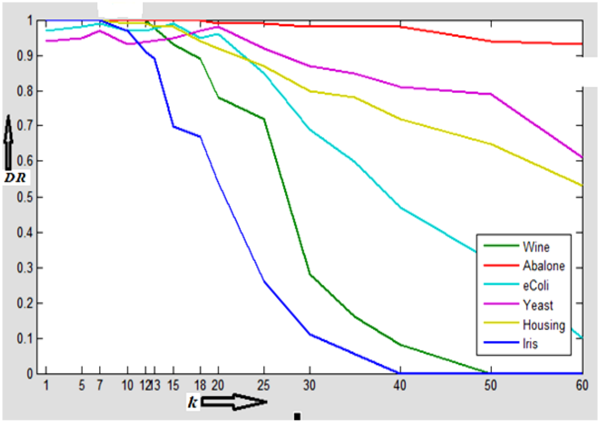

Figure 12. Illustrating the accuracy of outlier detection for different values of k in LSH-OD. It is useful for identification of threshold k range values, where it performs well

7 Conclusions and Future Work

We have developed a hybrid supervised outlier detection algorithm based on both distance and density based approach. The effectiveness of the algorithm results from the combined distance and density based outlier detection approaches. The distance based approach alone is able to detect outliers for datasets where objects are uniformly distributed among the data clusters. The weakness of distance based approach in detecting outliers for non-uniformly distributed datasets is compensated by density based approach by considering the local density around a candidate outlier object. Furthermore, the incorporation of entropy for outlier detection makes it more robust and sensitive than other existing outlier detection techniques. The computation of within-cluster entropy using Renyi's entropy measure has an advantage as it lends itself nicely to nonparametric estimation directly from data[25] and it considers how data are distributed within the cluster. Again, the proposed RODHA has a linear time complexity.

The algorithm is tested on synthetic and real-life datasets from UCI ML Repository. The detection performance of the algorithm is competing excellent than other existing algorithms. In the present work, the datasets on which the proposed technique is tested are of integer or real type. So, our work is undergoing to extend the algorithm to work on mixed type datasets. Apart from this, the performance of the algorithm will be tested over network intrusion datasets.

REFERENCES

[1] R. Barczynski. System outlier mining, 2010. http://www.xbow.com..

[2] M. Bawa, T. Condie, and P. Ganesan, Lsh forest: self-tuning indexes for similarity search. In Proc. of the 14th international conference onWorld Wide Web, pages 651-660, 2005.

[3] S. Bay and M. Schwabacher. Distance-based outliers in near linear time with randomization and a simple pruning rule. In Proceedings of the Ninth ACM SIGKDD, pages 29-38. Keleuven Press, 2003.

[4] M.M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In Proceedings of ACM SIGMOD on Management of Data, pages 386-395, 2000.

[5] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the 20th Annual Symposium on Computational Geometry, pages 253-262, New York, NY, USA, 2004. ACM.

[6] http://archive.ics.uci.edu/ml/datasets.

[7] W. Dong, Z.Wang, W. Josephson, M. Charikar, and K. Li. Modelling lsh for performance tuning. In Proc. of 17th International Conference on Information and Knowledge Management, pages 669-678, 2008.

[8] OnlineAvailable: http://en.wikipedia.org/wiki/Entropy/Information Theory.

[9] T. Fawcett. An introduction to roc analysis. Pattern Recognition Letters, 27(8):861-874, June 2006.

[10] P. Gogoi, D. Bhattacharyya, B. Borah, and J. Kalita. A survey of outlier detection methods in network anomaly identification. The Computer Journal (2011), 54(4):570-588, April 2011.

[11] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 500 Sansome Street, Suite 400, San Francisco, CA 94111, 2006.

[12] Z. He, X. Hu, and S. Deng. An optimal model for outlier detection in categorical Data. International Conference on Intelligent Computing, ICIC 2005, 3644:400-409, 2005.

[13] R. Jenssen, K. E. H. II, D. Erdogmus, J. C. Principle, and T. Eltoft. Clustering using Renyi's Entropy. In the proceedings of the International Joint Conference on Neural Networks, pages 523-528, Dept. of Electronics & Computer Engg., Florida University, Gainesville, FL, USA, 2003.

[14] H. Kim, G. H. Golub, and H. Park. Missing value estimation for dna microarray gene expression data: local least squares imputation. Bioinformatics, 21(2):187-198, 2005.

[15] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. VLDB Journal, 8: 237-253, 2000.

[16] H. D. K. Moonesinghe and P. N. Tan. Outrank: A graph-based outlier detection framework using random walk. International Journal on Artificial Intelligence Tools, 100(10):1-18, 2007.

[17] A. Renyi. On measures of entropy and information. In in Fourth Berkeley Symposium on Mathematical Statistics and Probability, pages 547-561, 1960.

[18] O. Troyanskaya,M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6):520-525, 2001.

[19] C. Varun, B. Arindam, and K. Vipin. Outlier detection - a survey. Technical report, Dept of CSE, University of Minnesota, USA, 2007.

[20] Y. Wang, S. Parthasarathy, and S. Tatikonda. Locality sensitive outlier detection: A ranking driven approach. Data Engineering (ICDE), 2011 IEEE 27th International Conference.

[21] A. Agrwal. Local Subspace based Outlier Detection. IC3 2009, CCIS 40, pp. 149–157, 2009.

[22] S. Seo. A Review and Comparison of Methods for Detecting Outliers in Univariate Data Sets, BS, Kyunghee University, 2002.

[23] A. Ghoting, M.E. Otey and S. Parthasarathy. LOADED: Link-based Outlier and Anomaly Detection in Evolving Data Sets. In Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM '04) 0-7695-2142-8/04.

[24] S. Wu and S. Wang. Information-theoretic Outlier Detection for Large-scale Categorical Data. IEEE Transactions on Knowledge and Data Engineering 2011.

[25] J. Principe, D. Xu, and J. Fisher, Unsupervised Adaptive Filtering, vol. 1, chapter 7 "Information Theoretic Learning", John Wiley & Sons, 2000.

[26] E. Parzen, "On the Estimation of a probability density function and the mode," *Ann. Math. Stat.*, vol. 32, pp. 1065–1076, 1962.