Outlier Detection using Cluster-Based Approach

Ms. Mayuri Anil Bhangare[1], Prof. J. R. Mankar[2]

[1] *PG Student, Department of Computer Engineering, KKWIEER, Nashik, Maharashtra, India*
[2] *Assistance Prof, Department of Computer Engineering, KKWIEER, Nashik, Maharashtra, India.*

ABSTRACT

Data mining has the crucial task of Outlier detection which aims to detect an outlier from given data set. The data is said to be an outlier which appears to have inconsistent observation with the remaining data. Outliers are generated because of improper measurements, data entry errors or data arriving from various sources than remaining data. Outlier detection is the technique which discovers such type of data from the given data set. Several techniques of outlier detection have been introduced which requires input parameter from the user such as distance threshold, density threshold, etc. The goal of this proposed work is to partition the input data set into the number of clusters using Unsupervised Extreme Learning Machine algorithm. Then the clusters are given as an input to the outlier detection methods namely cluster based outlier detection algorithm and outlier detection algorithm. The methods detects an outlier from each cluster. This work aims at studying cluster based outlier detection algorithm and outlier detection algorithm with different data sets. Also analyzing the performance of each method based on outlier detection accuracy.

Keyword: - *Clustering, Cluster-based outlier detection, Data mining, Outlier Detection.*

1. INTRODUCTION-

Outlier is a data which appears to have inconsistent observation with the remaining data and outlier detection is the technique which discovers such type of data from the given data set. Outlier is generated because of data entry errors, improper measurements or data arriving from various sources than rest of the data [13].

Outlier detection is an important task in data mining which aims to detect an outlier from given data set. Outlier detection is the first step towards obtaining a coherent analysis in many data-mining applications. The technique of outlier detection is used in many fields such as data cleansing, environment monitoring, criminal activities in e-commerce, clinical trials, network intrusion detection etc.

The cluster-based outlier detection the method works in two phases. In first phase the data set needs to be clustered using Unsupervised Extreme Learning Machine [12]. Unsupervised learning machine (US-ELM) deals with unlabeled data and performs clustering efficiently. US-ELM can be used for multicluster clustering for unlabeled data. In second phase the outliers are detected from each cluster.

Proposed system extends ELM to Unsupervised Extreme Learning Machine. It deals only with unlabeled data and also handles clustering task efficiently. Proposed system works in two phases where in first phase k-number of clusters is generated using US-ELM from input data set and in second phase an outlier is detected from each cluster using different methods viz. Outlier Detection Algorithm (ODA) and cluster based outlier detection algorithm for same data sets. Then the systems final output is the set of outliers. The main focus of this paper is on the analysis of the methods used for outlier detection. The performance of the system is measured in terms of outlier detection accuracy.

In cluster based outlier detection algorithm [14] all the clusters generated by US-ELM given as an input and the process is repeated for each cluster. The distance between centroid and each point is calculated and the points are sorted in descending order. For point p if the, find its k-nearest neighbor's (kNNs), the distance of point which is farthest from centroid in set of kNNs is considered as temporary distance. The point q is said to be an outlier if it follows conditions explained in following section in detail.

In ODA all the clusters generated by US-ELM are given as an input and then the radius of each cluster is calculated [10] .The pruning operation is performed in each cluster for the point whose distance from centroid is less than the radius of the cluster and other point remains unpruned. Once the pruning is over for all clusters the Local outlier Factor (LOF) for unpruned points is calculated which gives information about how much the point differs from its neighbor's. If the LOF of unpruned point is higher that means it is highly deviated from its neighbor's and the point is declare as an Outlier.

2. RELATED WORK

Huang et al. [1] introduced Extreme Learning machine (ELM) used for training Single Layer Feed Forward Network (SLFNs).The bias and parameters of SLFNs are randomly generated and ELM updates the output weights between hidden layer and output layer. ELM solves regularized least squared problem quicker than the Support Vector Machine's (SVM) quadratic programming problem. But ELM only works with labeled data.

D. Liu [2] extended ELM to the Semi-Supervised Extreme Learning Machine (SS-ELM) where the manifold regularization architecture was imported into the ELMs model to manage both unlabeled and labeled information. When the number of instances is higher than the number of neurons the SS-ELM and SS-ELM are work effectively. But SS-ELM is not able to achieve this because the data is not sufficient as compared to the number of hidden neurons.

J. Zhang [3] proposed co-training technique to train ELMs in SS-ELM. The labeled training sets grows progressively by transferring a subset of most positively judged unlabeled data at each iteration to the labeled set, and pseudo-labeled set is generated. This newly generated pseudo-labeled set is used to train ELMs regularly. The ELM's needs to train regularly in this algorithm, it makes effects on computational cost.

Statistical community [4, 5] is the first to study the problem of outlier and proposed model based outliers. They assumed that the data set follows some distribution or at least statistical estimates of unknown distribution parameters. An outlier is the data from data set that deviates from assumed distribution of data set. These model based approaches degrades their performance with high dimensional data set and arbitrary data set since there is no chance to have prior knowledge about distribution followed by these type of data set.

K. Li [6] proposed some model free outliers methods to overcome the drawback of model based outliers. There are two model free outliers detection approaches viz. Density based and Distance based. But these two model free outlier approaches required some input parameter to declare an object as an outlier e.g. distance threshold, number of objects nearest neighbor, density threshold etc.

Knorr and Ng [7-9] proposed another algorithm Nested-Loop (NL) to compute distance-based outlier. In this algorithm the buffer is partitioned into two halves viz. first array and second array. It copies data set into both arrays and computes the distance between each pair of objects. The count of neighbor is maintained for objects in first array. It stops counting neighbors of an object as soon as count of neighbors reaches to the D. Drawback of this algorithm is it takes high computation time. Typically nested loop algorithm requires $O(N^2)$ distance computations where N is number of objects in data set.

Angiulli et al. [11] proposed a method Detecting Outliers Pushing objects into an Index (DOLPHIN) which works with data sets resident to disk. It is easy to implement and also can work with any data type. It has I/O cost of successive reading two times the input data set file is inputted. Its performance is linear in time with respect to data set size since it performs similarity search without pre-indexing the whole data set. This method is improved further in efficient computations adopting spatial indexing by other researchers e.g. R-Trees, M-Trees etc. But these methods are sensitive to the dimensions.

3. SYSTEM ARCHITECTURE

The details of the overall system are described in this section. The Block diagram of the proposed system is shown in the Figure 1. It describes the overall working of the proposed system.

The proposed system works in two phase's viz. clustering and outlier detection. Before performing clustering input dataset is brought to the embedded space using Unsupervised Extreme Learning Machine (US-ELM) and then the clustering is performed in embedded space using k-means algorithm. After clustering, each cluster is given as an input to the outlier detection methods. There are two different methods used for outlier detection. These methods give set of outliers from input dataset as an output. The details of system architectural diagram are explained below.

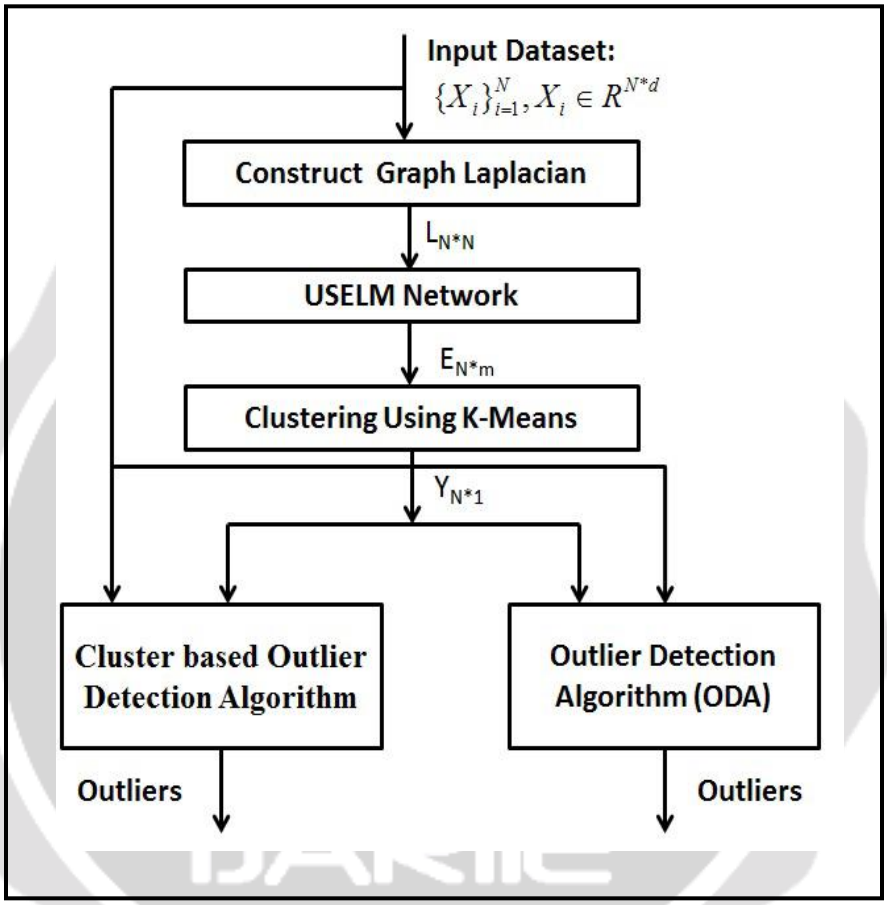

Figure 1: System Architectural Diagram

The proposed US-ELM processes unsupervised dataset. In unsupervised learning, the input data $\{X_i\}_{i=1 \text{ to } N}$, $X \epsilon R^{N*d}$ are unlabeled (N is the number of input instances) and our target is to find the basic structure of the original data. US-ELM algorithm map the input dataset to the embedded space $E \epsilon R^{N*m}$, where m is the dimensions of embedded space decided by us.

3.1 Graph Laplacian (L)

L= D – W, is the graph Laplacian of size N *N, Where W is Similarity or weight matrix of size N*N and the nonzero weights are computed by Eq (1.1) which is nothing but the Gaussian function.

$$W = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right) \quad \ldots\ldots\ldots (1.1)$$

D is Degree Matrix of size N *N. It is diagonal Matrix with diagonal matrix and computed by Eq (1.2).

$$D_{ii} = \sum_{j=1}^{N} W_{i,j} \qquad \ldots\ldots\ldots (1.2)$$

3.2 US-ELM Network

The US-ELM network follows Single Layer Feed Forward Networks (SLFNs) and it has three layers to map the input data to the embedding space namely input layer, hidden layer and output layer.

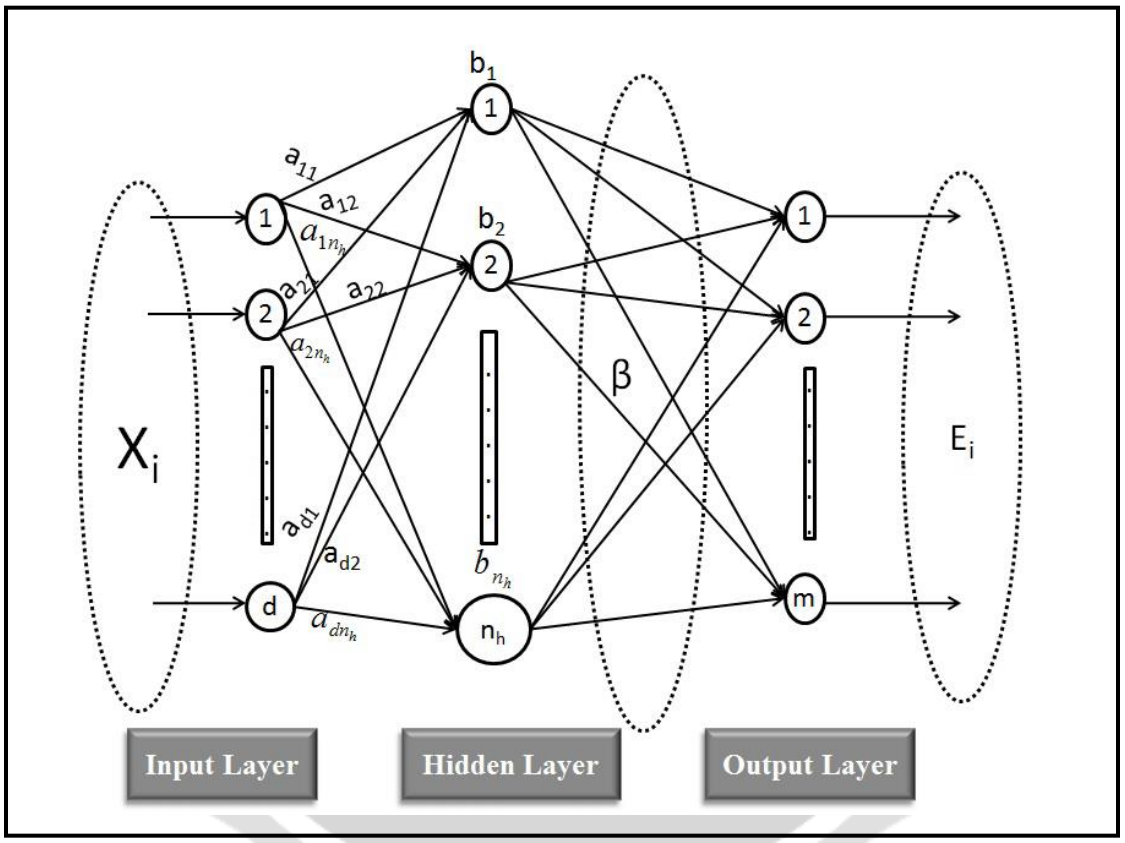

Figure 2 US-ELM Network Architecture

The Figure 2 shows architecture of US-ELM in which how the d-dimensional input is mapped to the m-dimensional embedded space shown. The outputs y_i and y_j of US-ELM are close to each other iff two samples of input dataset x_i and x_j are close to each other.

In the initial stage of US-ELM hidden layer is constructed using randomly generated fixed number of mapping neurons. The mapping function can be any linear piecewise continuous functions such as Sigmoid function by Eq (1.3) and Gaussian function by Eq (1.4).

$$g(x_i, \theta) = \frac{1}{1 + \exp(-(a_j{}^T x_i + b_j)} \qquad \ldots\ldots\ldots (1.3)$$

$$g(x_i;\theta) = \exp(-b_j \parallel x_i - a_j{}^T \parallel) \qquad \ldots\ldots\ldots\ldots \quad (1.4)$$

where θ = {a, b} are the parameters of the mapping function where a is vector which are input weights, b is scalar which is bias and ‖. ‖ denotes the Euclidean norm.

By using any continuous probability distribution e.g. the uniform distribution on (-1;1) the parameters of mapping functions i.e. a and b can be randomly generated is the main feature US-ELM.

The output of the hidden layer is denoted as H and it is given by,

$$H = \begin{pmatrix} g(x_1, a_1{}^T, b_1) & \ldots\ldots & g(x_1, a_{n_h}{}^T, b_{n_h}) \\ . & . & . \\ . & . & . \\ . & . & . \\ g(x_n, a_1{}^T, b_1) & \ldots\ldots & g(x_n, a_{n_h}{}^T, b_{n_h}) \end{pmatrix}$$

Then in second stage US-ELM aims to obtain the output weights by minimizing the cost function in eq (1.5). If an output layer of US-ELM has m number of nodes, then choosing the output weights b whose columns are the eigenvectors corresponding to the first m smallest eigenvalues is an optimal solution to objective function (1.5).

$$\min_{\beta \in R^{n_h * m}} \parallel \beta \parallel^2 + \lambda Tr(\beta^T H^T L H \beta)$$
$$s.t.\ (H\beta)^T H\beta = I_m \qquad \ldots\ldots\ldots\ldots \quad (1.5)$$

where Im : Identity matrix,
λ is the hyper parameter and it is selected from the exponential sequence $\{10^{-4}, 10^0 \ldots 10^4\}$ based on the clustering performance,
Tr(.) denotes the trace of a matrix ,
$\beta \in R^{n_h * m}$ the output weights that connect the hidden layer with the output layer.

An optimal solution to eq(1.5) is given by choosing b as the matrix whose columns are the eigenvectors corresponding to the first m smallest eigenvalues of generalized eigenvalue problem :
If $n_h \leq N$,

$$(I_{n_h} + \lambda H^T L H)v = \gamma H^T H v \qquad \ldots\ldots\ldots\ldots \quad (1.6)$$

$where\ A = I_{n_h} + \lambda H^T L H$,

$B = H^T H$,

$I_{n_h} = Identity\ Matrix,$

$\gamma_i = i^{th} smallest\ eigenvalue\ of\ eq(3.6),$

$v_i = correspondng\ eigenvecto\sigma,$

β *i.e. solution of output weights is given by*

$$\beta^* = \{\tilde{v}_2, \tilde{v}_3, ..., \tilde{v}_{m+1}\}$$

where $\tilde{v}_i = v_i / \| Hv_i \|, i = 2,3,..., n_{m+1}$ *are the normalized eigenvectors.*

If $n_h > N$,

$$(I_N + \lambda LHH^T)u = \gamma HH^T u \quad \ldots\ldots\ldots\ldots (1.7)$$

where $A = I_N + \lambda LHH^T$

$B = HH^T$,

I_N = *Identity Matrix,*

u_i = *correspondng eigenvector of eq*(3.7),

β *i.e. solution of output weights is given by*

$$\beta^* = \{\tilde{u}_2, \tilde{u}_3, ..., \tilde{u}_{m+1}\}$$

where $\tilde{u}_i = u_i / \| H^T Hu_i \|, i = 2,3,..., n_{m+1}$ *are the normalized eigenvectors.*

Embedding Matrix E_{N*m} as final output of US-ELM is computed by Eq (1.8) ,

$$E_{N*m} = H_{N*n_h} * \beta_{n_h*m} \quad \ldots\ldots\ldots\ldots (1.8)$$

33 K-Means Clustering

After the data is brought into the embedded space, now k-means algorithm for clustering is adopted. Each row of E i.e. E_i is treated as point to perform clustering. In such way N samples are clustered into k number of clusters.

Let Y_{N*1} be the output column vector of clustering task which consist of label vector with cluster index for all N samples from given data set.

3.4 Cluster Based Outlier Detection Algorithm

3.4.1 Distance between two points

Given dataset $\{X_i\}_{i=1 \text{ to } N}$, $X \epsilon R^{N*d}$, a point p is denoted by $p =< p[1], p[2], p[3],...., p[d] >$. The distance between two points p_1 and p_2 is given by Euclidean distance eq (1.9).

$$dist(p_1, p_2) = \sqrt{\sum_{i=1}^{d} (p_1[i] - p_2[i])^2} \quad \ldots\ldots\ldots\ldots \quad (1.9)$$

3.4.2 Centroid of cluster

Suppose that an input dataset X is clustered into k clusters $C_1, C_2, C_3 \ldots\ldots, C_n$ by US-ELM algorithm. Then the centroid of C_i.center of cluster C_i is computed by eq (1.10)

$$C_i.center = \frac{\sum_{p \in c_i} p[i]}{|C_i|} \quad \ldots\ldots\ldots\ldots \quad (1.10)$$

3.4.3 Algorithm

An input dataset $X \epsilon R^{N*d}$ is divided into n number of clusters viz. $C_1, C_2, C_3 \ldots\ldots, C_n$ by USELM and given as input to the outlier detection. Outlier detection is performed on each cluster.

For a cluster C, all the points in cluster C are sorted in ascending order of their distance from centroid. For any point p in sorted order of C, we scan all the points to search to its kNNs. Let $nn_k^{temp}(P)$ be the set of kNN points of p and $kdisp_{temp}(p)$ be the maximum distance from set $nn_k^{temp}(P)$ to the point p. The following method describes pruning method.

Theorem 1: For a point q in front of p, if dis (q, C.center) < dis (q, C.center) - $kdis_{temp}$ p), the points in front of q and q itself cannot be the kNNs of p.

Theorem 2: For a point q at the back of p, if dis (q, C.center) > dis (q, C.center) + $kdis_{temp}$ (p), the points at the back of q and q itself cannot be the kNNs of p.

3.5 Outlier Detection Algorithm (ODA)

An input dataset $X \epsilon R^{N*d}$ is divided into n number of clusters viz. $C_1, C_2, C_3 \ldots\ldots, C_n$ by USELM and given as input to the outlier detection. Outlier detection is performed on each cluster.

Radius pruning is implemented in this algorithm. Pruning step is used to remove the points which are very close to the centroid of the cluster. For this, radius of each cluster is calculated

3.5.1 Radius of Cluster

Cluster C having points each is of d-dimensions then the radius of cluster is computed by eq (1.11)

$$radius = \sqrt{\frac{\sum_{i=1}^{d} (p_1[i] - c.center[i])^2}{|C|}} \quad \ldots\ldots\ldots\ldots \quad (1.11)$$

3.5.2 Local Outlier Factor

Local outlier factor (LOF) for each point in the dataset gives the information about degree of outlier-ness. The outlier factor is said to be local means only a limited neighborhood of each point is taken into account. The LOF of an object is based on the single parameter of MinPts, which is the number of nearest neighbors used in defining the local neighborhood of the object.

(a) Distance between each two points

Calculate all the distances between each two data points of given dataset.

(b) k-distance of an object p

For any positive integer k, the k-distance of object p, denoted as k-distance(p), is defined as the distance d(p,o) between p and an object o ϵ D such that:

i. for at least k objects o'ϵ D|p it holds that d(p,o) ≤ d(p,o), and
ii. for at most k-1 objects o'ϵ D|p it holds that d(p,o') < d(p,o).

(c) k-distance neighborhood of an object p

Given the k-distance of p, the k-distance neighborhood of p contains every object whose distance from p is not greater than the k-distance i.e. $N_k(p) = o'|o'\epsilon D, dist(p,o) \leq dist_k(p)$.

(d) Reachability distance of an object p w.r.t. object o

Let k be a natural number. The reachability distance of object p with respect to object o is defined as $reach\text{-}dist_k(p,o)=\max\{k\text{-distance}(o),d(p,o)\}$.

(e) Local reachability density of an object p

The local reachability density of p is defined by eq (1.12).

$$lrd_{MinPts}(p) = 1 \Big/ \left[\frac{\sum_{o \in N_{MinPts}(p)} reach-dist_{MinOts}(p,o)}{|N_{MinPts}(p)|} \right] \quad \ldots\ldots\ldots\ldots (1.12)$$

The local reachability density of an object p is the inverse of the average reachability distance based on the MinPts nearest neighbors of p. Note that the local density can be ∞ if all the reachability distances in the summation are 0. This may occur for an object p if there are at least MinPts objects, different from p, but sharing the same spatial coordinates, i.e. if there are at least MinPts duplicates of p in the dataset. For simplicity, we will not handle this case explicitly but simply assume that there are no duplicates.

(f) Local Outlier factor of an object p

The local outlier factor of p is defined by eq (1.13).

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \quad \ldots\ldots\ldots\ldots (1.13)$$

The outlier factor of object p captures the degree to which we call p an outlier. It is the average of the ratio of the local reachability density of p and those of p's MinPts-nearest neighbors.

3.5.3 ODA Algorithm

ODA algorithm works in three steps.

(a) If the number of points in cluster is very small then the cluster is not passed to the radius pruning.
(b) Distance of each point of a cluster from centroid is computed. If the distance of a point is less than the radius of a cluster, the point is pruned.
(c) The Local Outlier Factor (LOF) is calculated for all the ponts which are remained unpruned from all clusters in (a) and (b). If the LOF **is** greater than the threshold then the point is declared as an output.

4. EXPERIMENTS

In order to evaluate the performance of the proposed system the datasets Iris, Wine and Spambase are collected from UCI repository. All these datasets have different number of instances as well as different number of attributes.

Table 1 shows the details of datasets used for the experiments. In order to conduct the experiment, initially US-ELM algorithm is applied to get number of clusters from input dataset. Then an outliers are detected from input dataset using both the techniques Cluster based outlier detection algorithm and ODA.

Table 1 Dataset Details

Dataset Name	No. of Instances	No. of attributes
IRIS	150	4
WINE	178	13
SPAMBASE	4601	57

The performance of outlier detection system is measured in terms of outlier detection accuracy i.e. how correctly outliers are detected. Percentage of accuracy is calculated using equation (1.14).

$$Accuracy = \frac{No.\ of\ correctly\ identified\ outliers}{Total\ no.\ of\ outliers} * 100 \quad \ldots\ldots\ldots\ldots (1.14)$$

The Accuracy of the system is shown in Table 2 and Table.3 for all the datasets mentioned in Table 1. The table 2 and Table 3 shows the accuracy for k=2 and k=4 respectively for both the methods cluster based outlier detection algorithm and ODA where k is used in finding k nearest neighbors of a point in both the algorithms.

Table 2 shows that the ODA has better accuracy than cluster based outlier detection algorithm. Also the dimension of the dataset majorly affects accuracy of both the algorithms. Similarly in Table 3.ODA has better accuracy than cluster based outlier detection algorithm.

Table 2 Outlier detection accuracy at k=2

Dataset Name	Cluster based outlier detection algorithm (in %)	ODA (in %)
IRIS	95	85
WINE	49	76
SPAMBASE	27	47

In ODA radius pruning is done before finding an outlier hence there are less chances of detecting normal data as an outlier where as in case of cluster based outlier detection algorithm system detects an outlier correctly but it also detects normal data as an outlier. Hence by observations it seems that ODA gives more accurate outliers as compared to cluster based outlier detection algorithm.

Table 3 Outlier detection accuracy at k=4

Dataset Name	Cluster based outlier detection algorithm (in %)	ODA (in %)
IRIS	79	90
WINE	67	82
SPAMBASE	19	58

5. CONCLUSION

Outlier detection is a significant problem with direct application in a wide variety of domains. An important observation with outlier detection is that it is not a well-formulated problem. Several approaches have been proposed to target a particular application domain. To detect cluster based outliers first input data set is clustered into number of clusters using Unsupervised Extreme Learning Machine (US-ELM) algorithm and then outlier is detected from each cluster by applying a cluster based outlier detection algorithm and the Outlier Detection Algorithm. The ODA algorithm with US-ELM gives better accuracy as compared to cluster based outlier detection algorithm with US-ELM for same set of datasets. The dimensions of datasets affect the performance of CBOD approach.

6. ACKNOWLEDGEMENT

The authors would like to thank Prof. Dr. S. S. Sane, Head of the Computer Department and Prof. Dr. K. N. Nandurkar, Principal of K.K.W.I.E.E.R., Nashik, for their kind support and suggestions. We would also like to thank all the staff members of Computer Engineering Department and our colleagues who knowingly or unknowingly helped us to complete this work.

7. REFERENCES

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in Proc. Int. Joint Conf. Neural Netw., vol. 2. 2004, pp. 985–990.

[2] L. Li, D. Liu, and J. Ouyang, "A new regularization classification method based on extreme learning machine in network data," J. Inf. Comput. Sci.,vol. 9, no. 12, pp. 3351–3363, 2012.

[3] K. Li, J. Zhang, H. Xu, S. Luo, and H. Li, "A semi-supervised extreme learning machine method based on co-training," J. Comput. Inf. Syst.,vol. 9, no. 1, pp. 207–214, 2013.

[4] Barnett, V., Lewis, T.: Outliers in Statistical Data. Wiley, New York (1994).

[5] Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley, New York (2005).

[6] He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recog. Lett. 24(9), 1641–1650 (2003).

[7] Knorr, E.M., Ng, R.T.: Algorithms for mining distance based outliers in large datasets. In: Proceedings of the

International Conference on Very Large Data Bases, pp. 392–403 (1998).

[8] Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. ACM SIGMOD Rec. 29(2), 427–438 (2000).

[9] Angiulli, F., Pizzuti, C.: Outlier mining in large high-dimensional data sets. IEEE Trans.

[10] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. ACM Sigmod Rec. 29(2), 93–104 (2000).

[11] Angiulli, F., Fassetti, F.: Very efficientmining of distance-based outliers. In: Proceedings of the Sixteenth ACMConference on Information and KnowledgeManagement, pp. 791–800 (2007).

[12] Angiulli, F., Fassetti, F.: Very efficientmining of distance-based outliers. In: Proceedings of the Sixteenth ACMConference on Information and KnowledgeManagement, pp. 791–800 (2007).

[13] Hawkins, D.M.: Identification of Outliers. Springer, New York (1980)

[14] X. Wang, D. Shen ,M. Bai, T. Nie ,Y. Kou ,G. Yu : Cluster-Based Outlier Detection Using Unsupervised Extreme Learning Machines.Springer International Publishing Switzerland 2016J. Cao et al. (eds.), Proceedings of ELM-2015 Volume 1, Proceedings in Adaptation, Learning and Optimization 6, DOI 10.1007/978-3-319-28397-5_11.