ISSN:0975-9646

An Improved Unsupervised Cluster based Hubness Technique for Outlier Detection in High dimensional data

R.Lakshmi Devi[#1], Dr.R.Amalraj[#2]

[#1]*ReseachScholar, Mother Teresa Women's University, Kodaikanal, India*
[#2]*Associate Professor, Dept. of Computer Science Sri Vasavi Arts College, Erode, India.*

***Abstract*-- Outlier detection in high dimensional data becomes an emerging technique in today's research in the area of data mining. It tries to find entities that are considerably unrelated, unique and inconsistent with respect to the common data in an input database. It faces various challenges because of the increase of dimensionality. Hubness has recently been developed as an important concept and acts as a characteristic for the increase of dimensionality connecting to nearest neighbors. Clustering also shows a vital role in handling high dimensional data and an important tool for outlier detection. This paper establishes a technique where the concept of hubness, especially the antihub (points with low hubness) algorithm is embedded in the resultant clusters obtained from clustering techniques such as K-means and Fuzzy C Means (FCM) to detect the outliers mainly to reduce the computation time. Further, the smaller clusters are treated as an outliers after applying clustering technique. So that they are all taken out before the antihub is applied, which further reduces the computation time. It compares the results of all the techniques by applying it on three different real data sets. The Experimental results demonstrate that when all five algorithms are compared, KCAntihubStage2 provides a significant reduction in computational time than the others and also provides better accuracy when the size of the data set is large. It is concluded that when the Antihub is applied into K-means, and the small clusters are removed, it outperforms well.**

Keywords: **Clustering Technique, Hubness, Outliers, Small Clusters, Unsupervised**

I. INTRODUCTION

An outlier is an observation which appears to be inconsistent with the remainder of that set of data. In data mining, detection of outliers is an important research area. Most of the applications which apply outlier detection are high dimensional. The sparsity of high dimensional data signifies that every point is an almost equally good outlier [1].

Outliers can be of three types such as point outliers, contextual outliers or collective outliers. Outlier (anomaly) detection refers to the process of finding patterns that do not conform to standard behavior. Outlier detection techniques can be classified into three different categories such as supervised, semi supervised and unsupervised based on the existence of the labels for outliers. The unsupervised outlier detection is more applicable, where dataset without the need of labels in the training set is given. The other techniques both require labeling data to produce the appropriate training set which is an expensive, time consuming and burdensome task [2]. In this paper, the proposed technique is applied to an unsupervised outlier detection.

The concept of hubness has recently become as an essential aspect of the increase of dimensionality related to nearest neighbors [3] and can be used in a standard methods used for detecting outliers. The hubness is explored in [4] as a new aspect of the increase of dimensionality and by examining the origin of hubness, authors show that it is an essential property of data distributions in high-dimensional data.

Clustering is a popular technique used to group similar data points or objects in groups or clusters [5]. Since clustering is an important tool for outlier analysis, it is focused along with hubness in this paper. This paper proposes a technique where the concept of hubness, mainly antihub algorithm is embedded in the resultant clusters obtained from clustering techniques such as K-means (KCAntihub) and Fuzzy C Means (FCAntihub) to detect the outliers. Small clusters are treated as an outliers and removed and then the antihub is applied to the remaining clusters (KCAntihubStage2, FCAntihubStage2). All are compared to find the efficient computation complexity among the all.

The rest of the paper is summarized as follows: Section 2 specifies an existing methods related to cluster based, and unsupervised outlier detection and Section 3 explains the proposed approach and its implications. Finally, Section 4 describes experimental evaluation with real datasets, and this chapter is concluded in Section 5.

II. RELATED WORK

In Recent research, various papers explored the influence of hubness in high-dimensional data on different data mining outlier detection tasks. Reverse nearest neighbors count is recognized in unsupervised distance-based outlier detection [3]. Outlier scoring based on Nk counts used in the ODIN method was reformulated and introduced here as an antihub which defines the outlier score of point x from data set D as a function of Nk(x) and explores the interplay of hubness and data sparsity. Outlier detection methods are implemented centered on the properties of antihubs (points with low hubness). The relationship between dimensionality, neighborhood size, and reverse neighbors are taken into account for the effectiveness.

Distance based method to deal with the problem of finding outliers for k dimensional data sets where k >=5 is

focused in paper [6]. Applying three algorithms such as index based, nested loop based, and cell based, authors come to the conclusion that cell based is for k<=4 and nested loop is the choice for k>=5 and also finds that there is no limit on the size of the dimensions.

A mostly used density based method is the local outlier factor (LOF) [7], which influenced many variations, e.g. LDOF (Local Distance-based Outlier Factor) approach [8], and LoOP (Local Outlier Probability) [9]. In many unsupervised outlier detection algorithms proposed, nearest-neighbor based algorithms appears to be the mostly used methods today [10, 11]. In this context, outliers are determined by their distances to their nearest neighbors.

Reference [12] explores an important feature of the curse concerning to the distribution of k-occurrences (the number of times a point appears among the k nearest neighbors of additional points in a data set) and shows that, as dimensionality increases, this distribution of data is skewed and hub points arise (points with very high k-occurrences).

There are many data mining cluster algorithms that detects data instance as an outlier which is situated far from other clusters. Among the unsupervised clustering algorithms, K-means algorithm is a widely used one and also considered as one of the top ten algorithms in data mining [13].

An approach is explained in [14] for the outlier detection of software measurement data using the K-means clustering method where the outlier which reduces the data quality is detected. [15] Presents an outlier detection approach based on the K-means clustering algorithm in order to separate the training data containing unlabeled flow of records into clusters of normal and abnormal traffic in network atmosphere. The resulting cluster centroids are used for the detection of anomalies.

Reference [16] proposes an efficient outlier detection method by applying K-means algorithm to recognize data instances which are not probable candidates for outliers by using the radius of each cluster and remove those data instances from the dataset. As an extension of the above discussion, the study in [17], [18] establishes an outlier detection method with the usage of K-means clustering for classifying abnormal and normal measures in a computer network.

As the boundaries between normal and outlier behavior cannot be well defined, outlier behavior in computer networks is very difficult to predict. The study of [19] establishes the idea of the fuzzy rough c-means (FRCM) to analyze clustering. FRCM gets the benefit of fuzzy set theory and rough set theory to predict the outlier in network intrusion detection.

Fuzzy c means clustering approach for outlier detection is presented in [20] and [21], [22] also utilizes the same clustering to remove noisy data and detect outlier. [23] Proposes a new approach, called FC-ANN based on ANN and fuzzy clustering to achieve higher detection rate, less false positive rate and stronger stability. An approach to identify the existence of breast cancer and calcification in mammograms using K-means and Fuzzy C-Means clustering is presented in [24]. The hybridization of fuzzy and neural computing system is very promising since they exactly tackle the situation associated with outliers. In [25], a Fuzzy min-max neural network is used for outlier detection.

The concept of hubness presented in [3] motivated to implement the proposed approach which is based on reverse nearest neighbors with antihub with the combination of clustering techniques and Euclidean distance as distance measure to find the neighbors.

III. PROPOSED APPROACH

A. *K-means Algorithm:*

K-means is a widely used unsupervised clustering algorithm, due to its simplicity and supports high dimensional data. Let X= {xi} i=1; . . . ; n be the set of n d-dimensional points to be clustered into a set of K clusters, C = {ck, k =1 . . . ; K}. K-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized.

Let μk be the mean of cluster ck. The squared error between μk and the points in cluster ck is defined as

$$J(C_k) = \sum_{x_i \in c_k} ||x_i - \mu_k||^2$$

The goal of K-means is to minimize the sum of the squared error over all K clusters,

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mu_k||^2$$

The main steps of K-means algorithm are as follows [26]:

1. Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.
2. Generate a new partition by assigning each pattern to its closest cluster center.
3. Compute new cluster centers.

B. *Fuzzy C Means Algorithm:*

Fuzzy C Means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. FCM is an unsupervised clustering algorithm that is applied to wide range of problems connected with clustering.

Consider a set of unlabeled patterns X={x1, x2…xN} $x_i \in R^f$ where N is the number of patterns and f is the dimension of features. FCM algorithm focuses on minimizing the value of an objective function which calculates the weighted within-group sum of squared errors as

$$Minimize \quad J_m(U,W) = \sum_{j=1}^{C} \sum_{i=1}^{N} (\mu_{ij})^m {d_{ij}}^2 \qquad (1)$$

Where

N: the number of patterns in X, C: the number of clusters, U: the membership function matrix; the elements of U are μij.

μij: the value of the membership function of the ith pattern belonging to the jth cluster.

dij: the distance from xi to wj, dij=||xi – wj (t) || where wj (t) denotes the cluster center of the jth cluster for the ith iteration

W: the cluster center vector, m: the exponent on μij; to control fuzziness or amount of clusters overlap.

The FCM algorithm focuses on minimizing Jm, subject to the following constraints on U:

μij ∈ [0, 1]. I=1…N and j=1…C (2)

$\sum_{j-1}^{C} \mu_{ij} = 1$, i=1…N (3)

$0<\sum_{i=1}^{N} \mu_{ij}<N$, j=1….C (4)

$$\mu_{ij}{}^{t} = \frac{1}{\sum_{i=1}^{C} (\frac{d_{ij}}{d_{il}})^{\frac{2}{(m-1)}}} \quad (5)$$

If dij =0 then μij=1 and μij=0 for l ≠j (6)

$$W_j^{(t)} = \frac{\sum_{i=1}^{N} (\mu_{ij}{}^{(t-1)})^m x_i}{\sum_{i=1}^{N} (\mu_{ij}{}^{(t-1)})^m} \quad (7)$$

The FCM algorithm starts with a set of initial cluster centers. Then it iterates the two updating functions at the ith iteration until the cluster centers are stable or the objective function in 1 converges to a local minimum. The algorithm consists of the following steps [27]:

Step1: Initialize the cluster center matrix, $W^{(0)}$

Step2: Initialize the membership matrix $U^{(0)}$ by using (5) and (6).

Step3: Increase t by one. Compute the new cluster center matrix $W^{(t)}$ by using (7)

Step4: Compute the new membership matrix $U^{(t)}$ by using functions (5) and (6).

Step5: if $||U^{(t)} - U^{(t-1)}|| < \varepsilon$ then stop, otherwise go to step 3.

C. Antihub:

Hubness is derived from the notion of k occurrences. Different data points occur in k nearest neighbor sets with increasingly unequal frequencies. When some points occur in many kNN sets, it is referred to as hubs, while others occur either very rarely or not at all, then they are referred to as antihubs. More specifically, hubness refers to an increasing skewness in the k occurrence distribution in high-dimensional data [28].

The concept of hubness has recently become as an essential aspect of the increase of dimensionality related to nearest neighbors. In summary, the emergence of antihubs is closely interrelated with outliers in high-dimensional data suggest that antihubs can be used as an alternative to standard outlier-detection methods. The development of antihub is closely associated with outliers by applying the hubness in to the resultant clusters which are constructed by the above mentioned clustering techniques.

D. Proposed algorithm:

The hybridization of clustering techniques such as K-means and FCM with hubness, especially antihub algorithm [3] is explored in this paper to improve the efficiency and to reduce the computational time. The block diagram of the proposed approach is given in Fig 1:

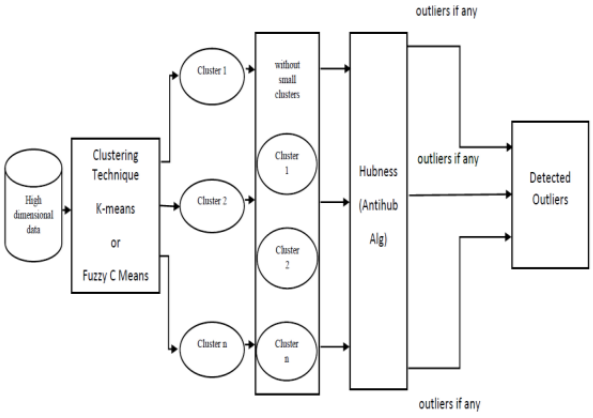

Fig. 1 Block Diagram of Proposed approach

The proposed system works in two phases. In the first phase, the clustering techniques such as K-means and FCM is applied in to the high dimensional data to obtain the groups of clusters and very small clusters are determined and considered as outliers [29]. In this case a small cluster is defined as a cluster with fewer points than half the average number of points in the k clusters and they are all taken out and then detect the outliers in the rest of clusters (if any).This pruning process really helps in reducing the computations. So that it reduces the computations and computation time required significantly to obtain outliers.

In the second phase the resultant clusters are taken into account for an input and for each cluster, antihub is applied where Nk value (reverse k-nearest neighbor count of x) is calculated with respect to distance and groups of clusters data, followed by outlier scores(si) are calculated for each data by means of monotone function. If the outlier score is less than outlier threshold, it is labeled as outlier. The proposed system can speed up the overall computation time and reduce the total number of computations to obtain outliers.

The basic structure of the proposed algorithm is as follows:

Input:

- High dimensional data set D = (x1, x2, . . . , xn), where xi ∈ Rd, for i ∈ {1, 2, . . . , n}
- No.of Clusters K ∈ {1, 2, . .N}

Output:

- Outliers

Steps:

1) Generate clusters.
   For each xi ∈ D where i ∈ (1, 2 . . . n)
   cd_j = Apply K-means / FCM (D, K) j ∈ (1, 2. . . K)
   // Groups of clustered data $cd_j = (cd_1, cd_2. \ . \ . \ cd_n)$, where cd_j ∈ D, for j ∈ {1, 2. . . K}
2) For each cd_j
   //Ordered clusters $cd_j=(x_1, x_2 \ldots x_{cn})$ $x_i \in cd_j$ where i ∈ 1, 2….c_n, $c_n=|cd_j|$
3) Determine small clusters and consider the points that belong to these clusters as outliers and prune them out.

If c_n < half the average number of points in the k clusters

Label the corresponding clusters as outliers and prune them out. Go to 4.

Else

4) Compute the outlier scores and find out outliers if it is less than outlier threshold.

for each i ∈ (1, 2….cn)

a. t:= $N_k(x_i)$ computed w.r.t. dist and clustered group of data cdj\xi

// N_k (x_i) is the reverse k-nearest neighbor count of x within D, D ⊂ Rd, xi ∈ D.

// Temporary variables: t ∈ R

//No. of neighbors k ∈ {1, 2 . . .},

//Distance measure dist (Euclidean distance).

b. s_i = f (t), where f: R → R is a monotone function.

c. If s_i > outlier threshold

Label it as an outlier.

The function f is 1/ (N_k(x) + 1), which assumes that the higher the score, the more the point is considered an outlier, and maps the scores to the (0, 1] range.

IV. EXPERIMENTAL EVALUATION

In this section, the effectiveness and behavior of the proposed approach is examined in terms of computation time and accuracy by applying it on three different real data sets obtained (wilt, aloi, and churn) against those algorithms. Wilt data set consists of image segments, generated by segmenting the pansharpened image with totally 4339 image segments. It involves 6 attributes. ALOI (Amsterdam Library of Object Images) dataset is a color image collection of 1, 00,000 small objects with 64 attributes. churn is a dataset with 1667 objects and 21 attributes. This section describes those experiments and their results.

Accuracy is the proportion of true results, either true positive or true negative, in a population. It measures the degree of veracity of a test on a condition. The terms that are used along with the description of accuracy are true positive (TP), true negative (TN), false negative (FN), and false positive (FP). Accuracy can be described as

Accuracy = (TN + TP)/ (TN+TP+FN+FP) = (No. of correct assessments)/No of all assessments)

TABLE I
THE COMPUTATION TIME OF ANTIHUB, KCANTIHUB, FCANTIHUB KCANTIHUBSTAGE2 AND FCANTIHUBSTAGE2 WHEN K=100 FOR ALOI, WILT AND CHURN

	Antihub(secs)	KCAntihub (secs)	FCAntihub (secs)	KCAntihub Stage2 (secs)	FCAntihub Stage2 (secs)
ALOI	3.0927	2.0537	2.9518	1.97	3.1264
WILT	9.495	2.701	6.405	1.8314	3.3697
CHURN	1.620	0.618	1.199	0.6103	1.1888

TABLE II
PERCENTAGE OF REDUCTION IN COMPUTATIONAL TIME FOR KCANTIHUB, FCANTIHUB, KCANTIHUBSTAGE2 AND FCANTIHUBSTAGE2 WHEN K=100

	KCAntihub in %	FCAntihub in %	KCAntihub Stage2 in %	FCAntihub Stage2 in %
ALOI	33.595	4.556	36.302	-1.090
WILT	71.557	32.539	80.712	64.510
CHURN	61.849	26.008	62.325	26.613
AVERAGE	55.667	21.034	59.779	30.011

Table I shows the computation time of all five algorithms for all the three datasets when k=100. Table II is derived from Table I to illustrate the percentage of reduction in computational time for KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 when k=100. From this it is well understood that significant reduction in an average of 59.779% in computation time occurs in KCAntihubStage2 for all the three datasets when compared with the existing Antihub system. It proves that KCAntihubStage2 outperforms well than the others.

Fig 2 demonstrates, that when we compare the computation time for the Antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 for all the three datasets, KCAntihubStage2 has a significant reduction in computation time for all three datasets and also proves that KCAntihubStage2 outperforms well among all the five.

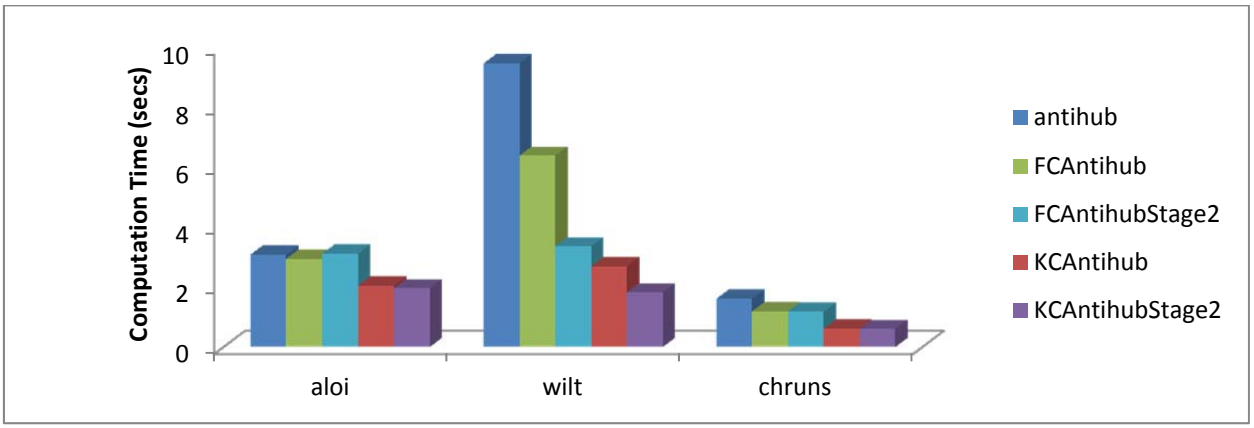

Fig 2 The computation time of antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 for ALOI, WILT and CHURN data sets

Table III shows that the Performance Accuracy of Antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 for ALOI, WILT and CHURN data sets when k=10, 50, 100 and 120. It also illustrates that accuracy in an average are at the same level for all the five algorithms for wilt and churn data sets. The accuracy of KCAntihubStage2 is little bit high for aloi data set where the number of objects is very high .

TABLE III
THE PERFORMANCE ACCURACY OF ANTIHUB, KCANTIHUB, FCANTIHUB KCANTIHUBSTAGE2 AND FCANTIHUBSTAGE2 FOR ALOI, WILT AND CHURN DATA SETS WHEN k=10, 50, 100 AND 120

	k Value	Antihub	KCAntihub	FCAntihub	KCAntihub Stage2	FCAntihub Stage2
ALOI	10	0.7625	0.7625	0.7625	0.8095	0.7564
	50	0.7603	0.7625	0.7625	0.8064	0.7547
	100	0.7595	0.7621	0.7625	0.806	0.7516
	120	0.7595	0.7621	0.7621	0.806	0.7503
	Average	0.76045	0.7623	0.7624	0.806975	0.75325
WILT	10	0.9776	0.9779	0.9765	0.9781	0.9772
	50	0.9829	0.9823	0.9827	0.9809	0.9804
	100	0.9829	0.9825	0.9829	0.9818	0.9827
	120	0.9829	0.9825	0.9832	0.982	0.9825
	Average	0.981575	0.9813	0.981325	0.9807	0.9807
CHURN	10	0.9904	0.991	0.9904	0.991	0.9904
	50	1	1	1	1	1
	100	1	1	1	1	1
	120	1	1	1	1	1
	Average	0.9976	0.99775	0.9976	0.99775	0.9976

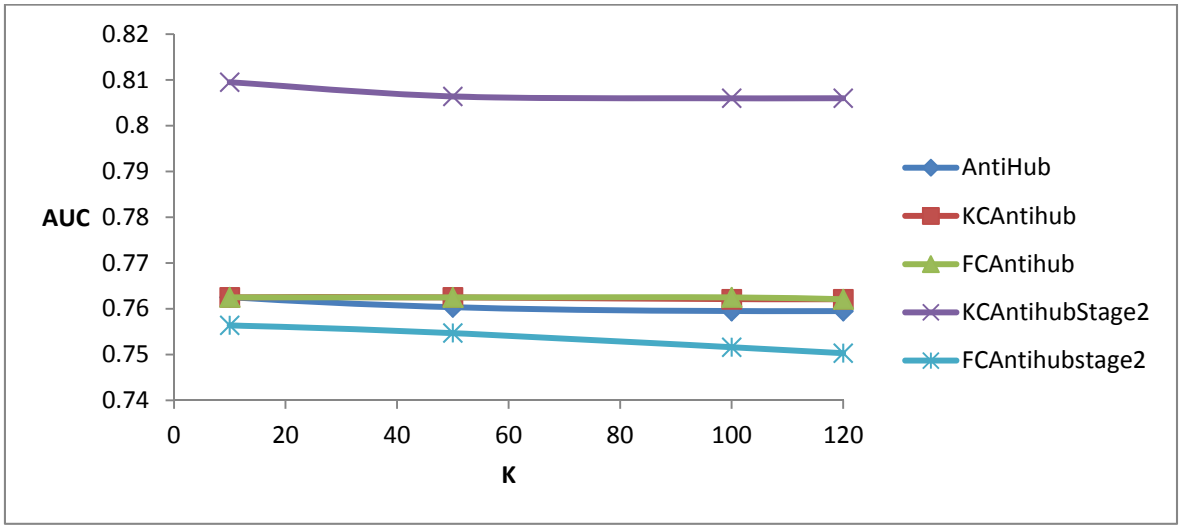

Fig 3 The performance accuracy of Antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 for ALOI dataset

Fig 3 shows that the performance accuracy of Antihub, KCAntihub, FCAntihub, and FCAntihubStage2 when using the dataset ALOI are at the same level when k=10, 50, 100, 120. KCAntihubStage2 accuracy is little bit more when it is compared with other four algorithms for the same k values for aloi data set where the number of objects is very high. Therefore while comparing the computation time of Antihub with other four in Table II, there is a significant reduction of 36.302% for KCAntihubStage2 is obtained with 0.806975 accuracy which is little bit high than the others when compared with the remaining algorithms in ALOI dataset.

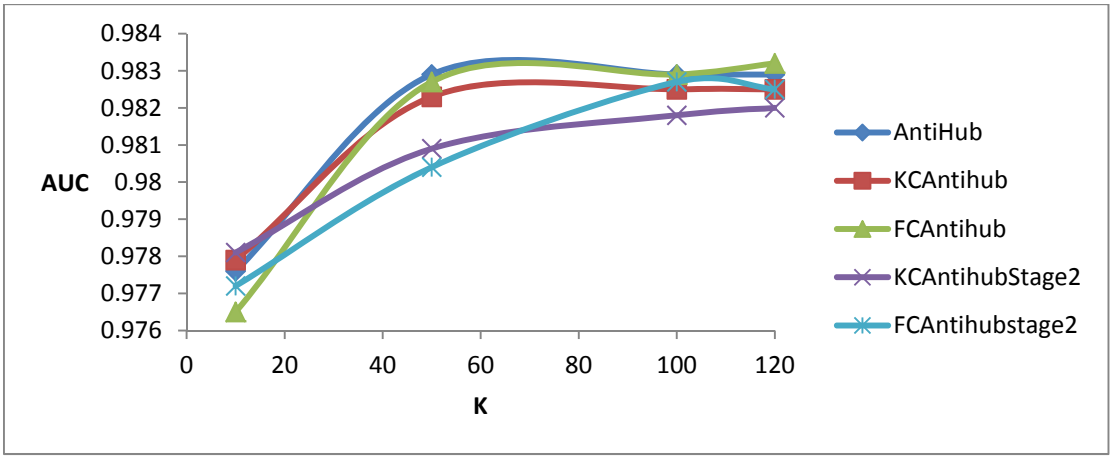

Fig 4 The performance accuracy of Antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 for WILT dataset

Fig 4 shows that the performance accuracy of Antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 when using the dataset WILT more or less are at the same level when k=10, 50, 100, 120. Therefore while comparing the computation time of Antihub with other four in Table II, there is a significant reduction of 80.712% for KCAntihubStage2 is obtained with the same level of accuracy for all five algorithms in WILT dataset.

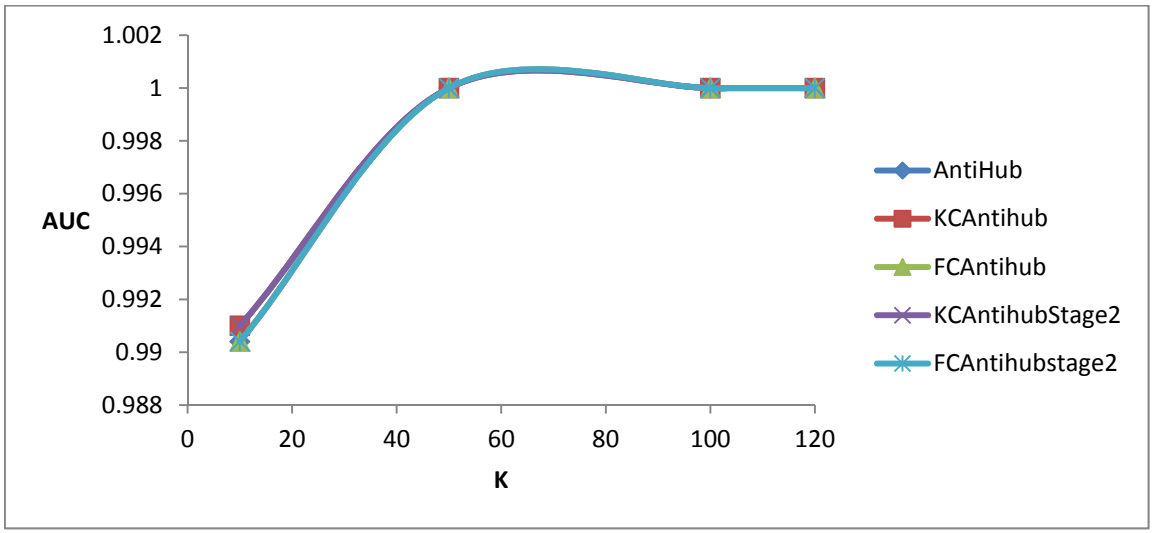

Fig 5 The performance accuracy of Antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 for CHURN dataset

Fig 5 shows that the performance accuracy of Antihub, KCAntihub, FCAntihub, KCAntihubStage2 and FCAntihubStage2 when using the dataset CHURN are at the same level when k=10, 50, 100, 120. Therefore while comparing the computation time of Antihub with other four in Table II, there is a significant reduction of 62.325% for KCAntihubStage2 is obtained with the same level of accuracy for all five algorithms in CHURN dataset.

V. CONCLUSION

This paper, presents an approach where hubness, especially antihub is applied to the clusters obtained from the clustering algorithms such as K-means and Fuzzy C Means for outlier detection to reduce computation time. Small clusters are then treated as outlier clusters and they are all taken out. The rest of outliers are then found (if any) in the remaining clusters by applying antihub. This process further reduced the computations and computation time. The performance of all algorithms are empirically compared in terms of computation time and accuracy by applying it into three different data sets and found that the KCAntihubStage2 provides a significant reduction in computational time than Antihub, FCAntihub, and FCAntihubStage2.It also provides better accuracy than the other algorithms when the size of the data set is very large. From this analysis it is concluded that when the antihub is applied to K-means and small clusters outliers are removed, this process, reduces the computation significantly and increases the performance accuracy. Finally it proves that KCAntihubStage2 outperforms well with the data set dimensionality than the other algorithms on identifying meaningful and interesting outliers.

REFERENCES

[1] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data" ACM SIGMOD RECORD February 2002.

[2] Amer, Mennatallah, and Slim Abdennadher. "Comparison of unsupervised anomaly detection techniques." Bachelor's Thesis (2011).

[3] Milo˜s Radovanovi´c, Alexandros Nanopoulos, and Mirjana Ivanovi´ "Reverse Nearest Neighbors in Unsupervised Distance-Based Outlier Detection" IEEE Transactions On Knowledge And Data Engineering, October 2014.

[4] M. Radovanovi´c, A. Nanopoulos, and M. Ivanovi´c, "Hubs in space: Popular nearest neighbors in high-dimensional data," J Mach Learn Res, vol. 11, pp. 2487–2531, 2010.

[5] Jain, Anil K. "Data clustering: 50 years beyond K-means." Pattern recognition letters 31.8 (2010): 651-666.

[6] Knorr, Edwin M., Raymond T. Ng, and Vladimir Tucakov. "Distance-based outliers: algorithms and applications." The VLDB Journal—the International Journal on Very Large Data Bases 8.3-4 (2000): 237-253.

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

[8] Zhang, Ke, Marcus Hutter, and Huidong Jin. "A new local distance-based outlier detection approach for scattered real-world data." Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, 2009. 813-822.

[9] Kriegel, Hans-Peter, et al. "LoOP: local outlier probabilities." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009

[10] Amer, Mennatallah, and Slim Abdennadher. "Comparison of unsupervised anomaly detection techniques." Bachelor's Thesis (2011).

[11] Amer, Mennatallah, and Markus Goldstein. "Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer." Proc. of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012). 2012.

[12] Radovanović, Miloš, Alexandros Nanopoulos, and Mirjana Ivanović. "Nearest neighbors in high-dimensional data: The emergence and influence of hubs." Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009

[13] XindongWu and et. al., 2008. Top 10 Algorithms in Data Mining, Journal of Knowledge and Information Systems, vol. 14. Issues 1-37. DOI: 10.1007/s10115-007-0114-2.

[14] Yoon, Kyung-A., Oh-Sung Kwon, and Doo-Hwan Bae. "An approach to outlier detection of software measurement data using the k-means clustering method." Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on. IEEE, 2007.

[15] Münz, Gerhard, Sa Li, and Georg Carle. "Traffic anomaly detection using k-means clustering." GI/ITG Workshop MMBnet. 2007.

[16] Pamula, Rajendra, Jatindra Kumar Deka, and Sukumar Nandi. "An outlier detection method based on clustering." Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on. IEEE, 2011.

[17] Muniyandi, Amuthan Prabakar, R.Rajeswari, and R.Rajaram. "Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree algorithm." Procedia Engineering 30 (2012): 174-182.

[18] Sharma, Sanjay Kumar, et al. "An improved network intrusion detection technique based on k-means clustering via Naïve bayes classification." Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on. IEEE, 2012.

[19] Chimphlee, Witcha, et al. "Anomaly-based intrusion detection using fuzzy rough clustering." Hybrid Information Technology, 2006. ICHIT'06. International Conference on. Vol. 1. IEEE, 2006.

[20] Al-Zoubi, Moh'D. Belal, Al-Dahoud Ali, and Abdelfatah A.Yahya. "Fuzzy clustering-based approach for outlier detection." Proceeding ACE 10 (2008): 192-197.

[21] Shahi, Ahmad, Rodziah Binti Atan, and Md Nasir Sulaiman. "Detecting effectiveness of outliers and noisy data on fuzzy system using FCM." Eur J Sci Res 36 (2009): 627-638.

[22] Kaur, Prabhjot, and Anjana Gosain. "Density-oriented approach to identify outliers and get noiseless clusters in Fuzzy C—Means." Fuzzy Systems (FUZZ), 2010 IEEE International Conference on. IEEE, 2010.

[23] Wang, Gang, et al. "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering." Expert Systems with Applications 37.9 (2010): 6225-6232.

[24] Singh, Nalini, Ambarish G. Mohapatra, and Gurukalyan Kanungo. "Breast cancer mass detection in mammograms using K-means and fuzzy C-means clustering." International Journal of Computer Applications (0975–8887) 22.2 (2011).

[25] Upasani, Nilam, and Hari Om. "Evolving Fuzzy Min-max Neural Network for Outlier Detection." Procedia Computer Science 45 (2015): 753-761.

[26] Jain, Anil K. "Data clustering: 50 years beyond K-means." Pattern recognition letters 31.8 (2010): 651-666.

[27] Hung, Ming-Chuan, and Don-Lin Yang. "An efficient fuzzy c-means clustering algorithm." Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE, 2001.

[28] N.Toma˜sev, R. Brehar, D. Mladeni´c, and S. Nedevschi, "The influence of hubness on nearest-neighbor methods in object recognition," in Proc. 7th IEEE Int. Conf. on Intelligent Computer Communication and Processing (ICCP), 2011, pp. 367–374.

[29] Loureiro, A., L. Torgo and C. Soares, 2004.Outlier Detection using Clustering Methods: a Data Cleaning Application, in Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector. Bonn, Germany