International Journal of Engineering & Technology, 7 (4.7) (2018) 292-296

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Distributed Mining of Outliers from Large Multi-Dimensional Databases

K. Ashesh[1]*, Dr. G. Appa Rao[2]

[1]Research Scholar, Dept. of Computer Science Engineering, GITAM University, Visakhapatnam, India
[2]Research Scholar, Dept. of Computer Science Engineering, GITAM University, Visakhapatnam, India
**Corresponding author E-mail: imasheshk@gmail.com*

Abstract

A data point is given dataset is considered to be outlier when it is not distant to all its nearest neighbours. Obviously it is based on distance measure. However, in distributed environments it is challenging to detect outliers. Many approaches to mine outliers such environments came into existence. However, a faster and more efficient way is desired. In this paper we employ a novel index tree which is hierarchical in nature. Its hierarchical structure paves way for space pruning while its clustering property helps in faster search of finding neighbours of a given data point. Its time complexity is linear to the size of dataset and its dimensions. On top of the hierarchical tree (Hi-tree) nearest neighbour search avoids unnecessary computations besides pruning unpromising points. An algorithm by name Distributed Mining of Outliers using Hi-tree (DMOH) is proposed. The index tree can be exploited with parallel processing phenomenon. We built a prototype application to demonstrate proof of the concept. Our empirical study revealed the efficiency of the proposed algorithm on top of Hi-tree.

Keywords: *Distributed outlier detection, outlier detection, hierarchical index tree.*

1. Introduction

Outliers are the abnormal data points in a dataset. They often provide required intelligence to make decisions. Outlier detection has many real time applications including fraud detection and intrusion detection to mention few [1]. In many real time applications, data may be in different networked machines. In such environments (distributed environments) data needs to be processed in different nodes as explored in [2] and [7]. However, it is very challenging to mine or discover outliers in distributed environments due to the dynamic nature of the environment. Many distance based outlier detection methods came into existence. The reason behind this is that distance is one of the simple metrics that can be used to know how an outlier is completely different from its nearest neighbours. In other words an object is considered as an outlier when it deviates significantly from its assumed distribution.

In the literature many approaches are found to detect outliers. The concept of natural neighbour [3], fast distributed outlier detection [2] and unsupervised outlier removal [24]. The existing systems found in the literature have different approaches in mining outliers. However, it is felt that a novel approach in finding outliers that reduces time and space complexity was desired. In this paper we proposed a framework for distributed outlier detection. It has an underlying hierarchical index structure that supports finding nearest neighbours and ability to prune search space to reduce time and space complexity. Thus the proposed system is more efficient in mining outliers in distributed environments. Different parameters like number of worker nodes are used to evaluate the performance of the proposed algorithm. Time complexity and data transfer are the two performance evaluation parameters used in this paper. Our contributions are as follows.

- We proposed a framework which facilitates number of worker nodes to operate on available data and produce local outliers. Then the local outliers are merged into global outliers.
- We proposed an algorithm named Distributed Mining of Outliers using Hi-tree (DMOH) for efficient detection of outliers in distributed environment.
- We built a prototype application to demonstrate proof of the concept. The application shows the ability of the proposed algorithm in mining outliers. It is a distributed application made up of RMI technology. It helps users to run different functions remotely and mine outliers from databases in distributed environments.

The remainder of the paper is structured as follows. Review of literature is presented in Section 2. The proposed methodology is provided in Section 3. Section 4 presents experimental results and the conclusions and directions for future work are provided in Section 5.

2. Related Work

This section provides review of literature on outlier detection. The notion of Local Outlier Factor (LOF) is employed to represent the likelihood of an object being an outlier. The problem with local outliers is that it consumes more resources for nearest neighbour search. Otey et al. [2] on the other hand proposed a mechanism for faster distributed outlier detection from datasets that do have multiple-attributes that exhibit heterogeneity in the kind of data. The notion of Natural Outlier Factor (NOF) is used by Huang et al. [3] for detecting non-parameter outliers. Campello et al. [4] on the other hand explored the process of hierarchical density estimates for outlier detection and other utilities. They used normalized score of outlierness in order to have better detection f outliers.

Bhuyan et al. [5] proposed a multi-step outlier detection mechanism that operates on network-wide traffic in order to find out outliers.

Rahmani et al. [6] employed Principal Component Analysis (PCA) for randomized space recovery and the concept of data sketching for detecting outliers in the high dimensional data matrices. Folino and Sabatino [7] explored collaborative filtering approaches in distributed environments in order to have an efficient detection of intrusions. Chu et al. [8] focused on the concept of identification and cleaning of dirty data. They studied and presented taxonomy of dirty data in the literature. Various anomaly detection techniques in the distributed environments are explored by Ahmed et al. [9]. Sarno et al. [10] on the other hand investigated a hybrid approach in association rule processing for detecting fraudulent transactions. They used event log for processing mining in order to have required business intelligence (BI). Nech et al. [11] on the other hand explored outlier detection in the image processing domain especially in the area of face recognition.

Shahid et al. [12] proposed a mechanism to detect outliers based on one-class SVM (Support Vector Machines) employed in Wireless Sensor Network (WSN). Similar kind of work is made in [13]. Schubert et al. [14] studied the challenges related to uncertain data and employed a method to have clustering of such data. Data mining methods are explored in [15] for cyber security intrusion detection based on the concept of finding outliers. Jia et al. [16] used anomaly detection thorough outliers and found performance degradation in wind turbines. Pack et al. [17] on the other hand explored visualization based approaches to find outliers. Megahed et al. [18] focused on the statistical process monitoring in the big data perspective for finding abnormal data. Begum et al. [19] proposed a methodology for Dynamic Time Wrapping (DTW) in order to have a better pruning strategy in clustering process. Guha et al. [20] explored random forests related to data mining for finding outliers in the data that is streamed continuously.

Table 1: Notations used in the proposed methodology

Notation	Description
$e.den$	the node density of e
$e.num$	the number of points in e
E	leaf node
I	level
s	Subspace
P	point
$dis_{in}(p,s)$	the minimum distance between p and the side of s
$s.\max$	Maximum point
$s.\min$	Minimum point

Qin et al. [21] explored data-centric approach in Internet of Things (IoT) in order to have robust detection of peculiar events. Rekatsinas et al. [22] proposed a holistic approach for data repairs that helps in improving utility of data. The concept of time-series clustering approach is employed in [23] for detection of outliers and concept drifts. Xia et al. [24] proposed an unsupervised machine learning model for discriminating and removing outliers. Jiang et al. [25] studied the graph mining approach to understand the happenings in the distributed environment in a network. From the literature it is understood that there are many attempts to find outliers in distributed environment. However, a distributed framework that can mine outliers more efficiently is desired. This paper proposed the same in order to have efficient detection of outliers in distributed environment.

3. Proposed Methodology

We proposed a methodology that is used to mine outliers in distributed environments. It acts on multi-dimensional data. The framework takes data from databases and performs pre-processing prior to mining outliers in distributed environment. The pre-processing is a simple mechanism of making the data ready for processing. Once pre-processing is completed, the data is partitioned into multiple portions and given to different nodes in distributed environment. The nodes perform the desired computations and discover outliers. Then the final outliers are formed by merging local outliers obtained from each node. Then the proposed outlier detection mechanism is subjected to evaluation to know its effectiveness.

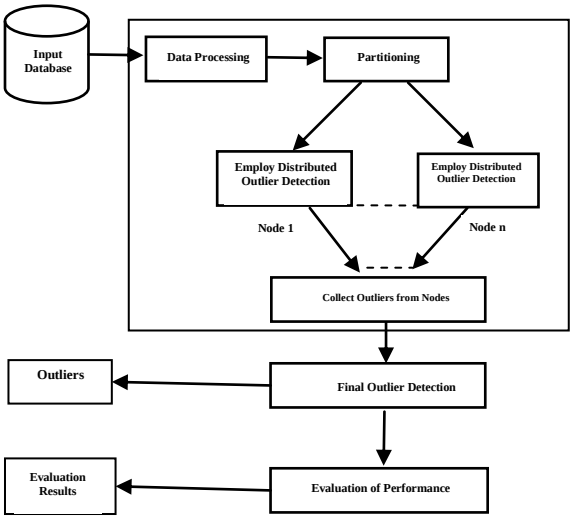

Figure 1: Proposed methodology for detecting outliers in distributed environment

As presented in Figure 1, the outcomes of the proposed methodology are outliers mined in distributed environment and the results of evaluation. An underling data structure known as hierarchical tree (Hi-tree) is built to represent data before actual processing. This data structure has two important features. The first one is it supports easy means of finding neighbours and the second one is that it supports pruning with the help of its spatial filtering ability. With respect to node density the tree's node density can be computed as in Eq. (1). The notations used in equations are found in Table 1.

$$e.den = \frac{e.num}{2^{-td}} \tag{1}$$

A node at the leaf with low-density assumes usually larger weight. Thus any greedy method can scan leaf node and find top-k nodes that are having smallest node densities. Minimum distance between a point p and the sub space denoted by its minimum point such as s.min is computed as in Eq. (2).

$$dis_{notin}(p,s) = \sqrt{\sum_{i=1}^{d} x_i^2} \tag{2}$$

Where the x_i is computed as in Eq. (3).

$$x_i = \begin{cases} p[i] - s.\max[i], if\ p[i] > s.\max[i], \\ s.\min[i] - p[i], if\ p[i] < s.\min[i], \\ 0, \quad Otherwise. \end{cases} \tag{3}$$

Where s.min and s.max are minimum and maximum distance points in given subspace denoted as s. If the two conditions are not satisfied, it results in zero. In the same fashion, the minimum dis-

tance between a side of subspace and a point is computed as in Eq. (4).

$$dis_{in}(p,s) = \min_{i=1}^{d} \{min\{p[i] - s.\min[i], s.\max[i] - p[i]\}\}$$

Based on the distance computations, the proposed algorithm performs its outlier detection activities and produce local outliers at each node. Then the outputs of local outlier detection are merged to form final outliers.

3.1 Distributed Mining of Outliers using Hi-tree (DMOH) Algorithm

This algorithm performs outlier detection locally and then the outputs are merged to form final outliers.

Algorithm: Distributed Mining of Outliers using Hi-tree (AMOH)
Inputs : Multi-dimensional large database D, number of nodes N
Outputs : Outliers and evaluation results

1. Initialize local outlier vector L
2. Initialize global outlier vector G
3. Partition data based on number of nodes
4. For each node n in N

At Each Node

5. For each row in Dn
6. Extract local outliers into L
7. End For
8. Add L to G
9. End For
10. Performance Evaluation
11. Return G and Performance Results

Algorithm 1: AMOH algorithm

The algorithm takes care of local outlier detection at each node and the outliers are used in later stage to merge with global outliers. The process is done parallel at each node and finally generates global outliers and also evaluation results. The following section shows the evaluation results in terms of time taken (time complexity), space complexity and so on.

4. Experimental Results

Experiments are made with the prototype application which demonstrates proof of the concept. The application is distributed in nature and built using Java RMI technology. It runs in multiple machines as per the framework and produces local outliers before merging them into global outliers.

Table 2: Shows time cost required by different algorithms

No. of Worker Nodes (m)	Time Cost (seconds)		
	DSS	DBOZ	Proposed
10	2000	600	500
15	1400	400	300
20	1100	300	200
25	900	200	100
30	800	100	50

The time taken for the proposed algorithm and other algorithms in seconds against different number of worker nodes is presented.

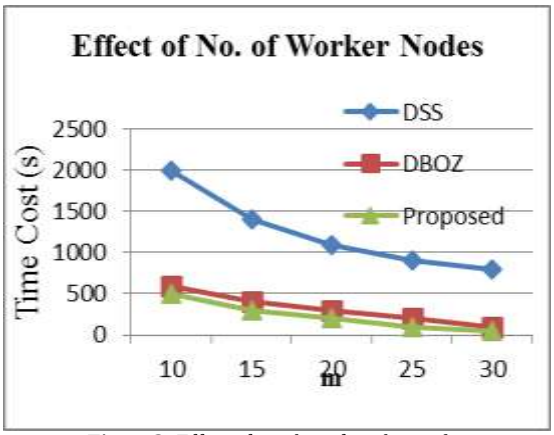

Figure 2: Effect of number of worker nodes

As shown in Figure 2, it is evident that the number of worker nodes is presented in horizontal axis while the vertical axis showed time cost in seconds. The results revealed that the number of nodes used for processing in distributed environment has its impact on response time. Another observation is that the proposed algorithm outperformed other existing algorithms.

Table 3: Shows ADT (MB) of different algorithms

No. of Worker Nodes	ADT (MB)		
	DSS	DBOZ	Proposed
10	400	50	40
15	500	50	40
20	600	50	40
25	700	60	50
30	800	70	60

As shown in Table 3, the ADT (MB) of the proposed algorithm and other algorithms against different number of worker nodes is presented.

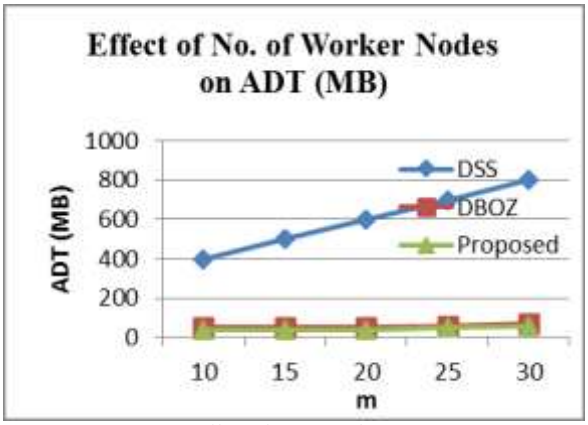

Figure 3: Effect of number of worker nodes

As shown in Figure 3, it is evident that the number of worker nodes is presented in horizontal axis while the vertical axis showed ADT (MB). The results revealed that the number of nodes used for processing in distributed environment has its impact on ADT (MB). Another observation is that the proposed algorithm outperformed other existing algorithms.

Table 4: Shows time cost required by different algorithms when n value is changed

Value of Parameter n	Time Cost (seconds)		
	DSS	DBOZ	Proposed
10	900	250	220
15	1000	270	250
20	1100	290	270
25	1200	310	290
30	1300	330	310

As shown in Table 4, the time cost required by the proposed algorithm and other algorithms in seconds against different value for parameter n is presented.

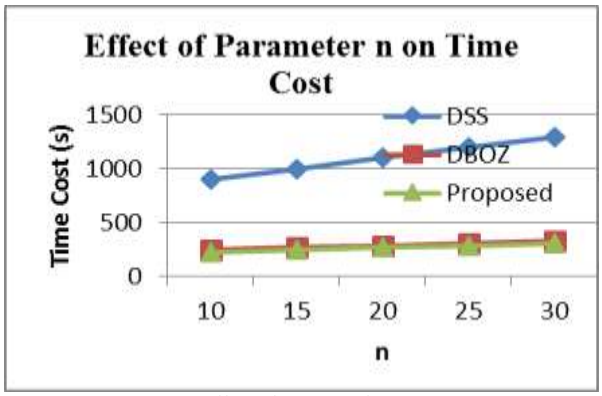

Figure 4: Effect of number of worker nodes

As shown in Figure 4, it is evident that the value of parameter n is presented in horizontal axis while the vertical axis showed time cost. The results revealed that the value of n used for processing in distributed environment has its impact on response time. Another observation is that the proposed algorithm outperformed other existing algorithms.

Table 5: Shows ADT (MB) required by different algorithms

Value for Parameter n	ADT (MB)		
	DSS	DBOZ	Proposed
10	500	40	30
15	550	40	30
20	600	50	40
25	650	60	50
30	700	70	60

As shown in Table 5, the ADT (MB) of the proposed algorithm and other algorithms against different number of worker nodes is presented.

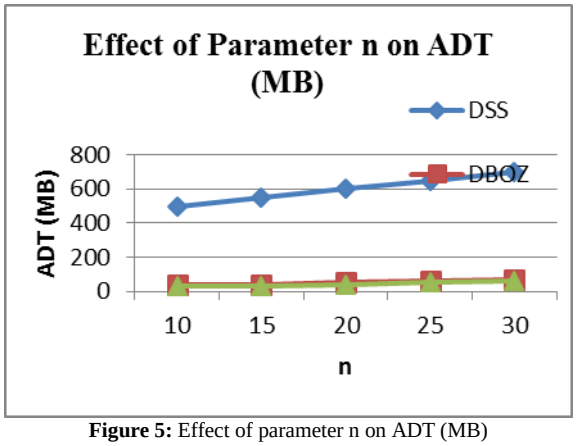

Figure 5: Effect of parameter n on ADT (MB)

As shown in Figure 5, it is evident that the parameter n is presented in horizontal axis while the vertical axis showed ADT (MB). The results revealed that the parameter n used for processing in distributed environment has its impact on ADT (MB). Another observation is that the proposed algorithm outperformed other existing algorithms.

Table 6: Shows time cost required by different algorithms based on parameter k

Value of Parameter k	Time Cost (seconds)		
	DSS	DBOZ	Proposed
10	900	220	200
15	1000	240	220
20	1100	260	240
25	1200	280	260
30	1300	300	280

As shown in Table 6, the time cost of the proposed algorithm and other algorithms against different values of parameter k is presented.

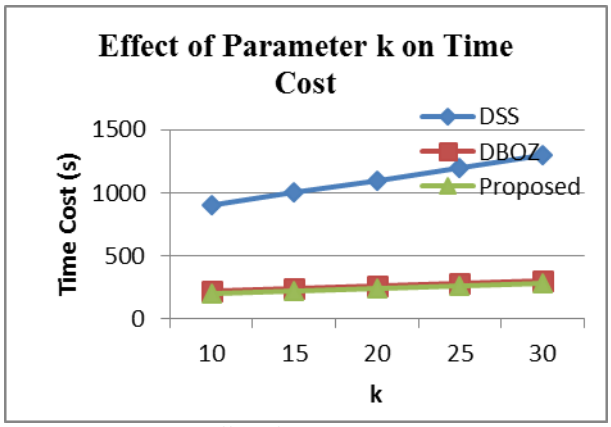

Figure 6: Effect of parameter k on time cost

As shown in Figure 6, it is evident that the parameter k is presented in horizontal axis while the vertical axis showed time cost. The results revealed that the parameter k used for processing in distributed environment has its impact on response time. Another observation is that the proposed algorithm outperformed other existing algorithms.

Table 7: Shows ADT (MB) of different algorithms based on parameter k

Value of Parameter k	ADT (MB)		
	DSS	DBOZ	Proposed
0	400	50	40
100	500	55	45
200	600	60	50
400	700	65	55
800	800	70	60

As shown in Table 7, the ADT (MB) of the proposed algorithm and other algorithms against different values of parameter k is presented.

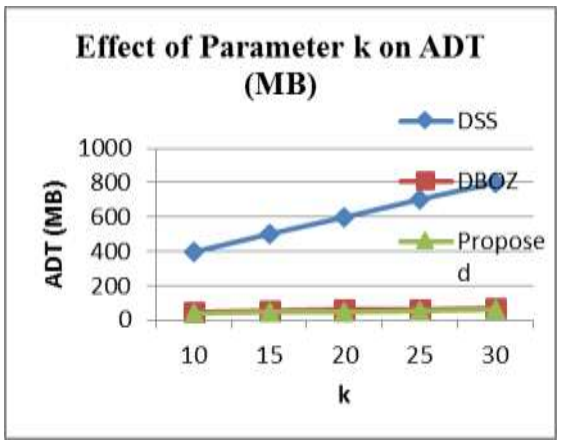

Figure 7: Effect of parameter k on ADT (MB)

As shown in Figure 7, it is evident that the parameter k is presented in horizontal axis while the vertical axis showed ADT (MB). The results revealed that the parameter k used for processing in distributed environment has its impact on ADT (MB). Another observation is that the proposed algorithm outperformed other existing algorithms.

5. Conclusions and Future Work

In this paper we studied the problem of mining outliers in distributed environments. We proposed a framework to have distributed outlier detection. The framework employs multiple nodes to process local outliers and then the local outliers merged to form global outliers. A data structure known as hierarchical index tree (Hi-tree) is used to achieve this. It has two important features. The first feature is it has property of clustering which can search for nearest neighbours easily. The second one is ability to prune search space to reduce time and space complexity. An algorithm by name Distributed Mining of Outliers using Hi-tree (DMOH) is proposed. A prototype application is implemented to demonstrate proof of the concept. The experimental results with different parameters are evaluated with time cost and ADT (MB). The results revealed that the proposed algorithm outperforms other state of the art algorithms. In future we intend to improve the algorithm further to work with live data streams and find outliers.

References

[1] Wen Jin Anthony K. H. Tung Jiawei Ha. (2001). Mining Top Local Outliers in Large Databases. *IEEE*, p1-6.

[2] Matthew Eric Otey, Amol Ghoting and Srinivasan Parthasarathy. (2005). Fast Distributed Outlier Detection in Mixed-Attribute Data Sets. *IEEE*, p1-33.

[3] Jinlong Huang, QingshengZhu , Lijun Yang, Ji Feng. (2016). A non-parameter outlier detection algorithm based on Natural Neighbor. *Elsevier*, p1-3.

[4] RICARDO J. G. B. CAMPELLO, DAVOUD MOULAVI,ARTHUR ZIMEK,JORG SANDER. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 10 (1), p1-52.

[5] Monowar H. Bhuyana, D.K. Bhattacharyya b , J.K. Kalitac. (2016). A multi-step outlier based anomaly detection approach to network wide traffic. *Elsevier*, p1-29.

[6] Mostafa Rahmani,and George K. Atia. (2017). Randomized Robust Subspace Recovery for High Dimensional Data Matrices. *IEEE*,p1-14.

[7] Gianluigi Folinon , Pietro Sabatino. (2016). Ensemble based collaborative and distributed intrusion detection systems A survey. *Journal of Network and Computer Applications*, p1-16.

[8] Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, Jiannan Wang. (2016). Data Cleaning Overview and Emerging Challenges. *IEEE*, p1-6.

[9] Mohiuddin Ahmed, Abdun Naser Mahmood, Jiankun Hu. (2016). A survey of network anomaly detection techniques. *Elsevier*, p1-13.

[10] RiyanartoSarno, RahadianDustrialDewandono, Tohari Ahmad, Mohammad Farid Naufal and Fernandes Sinaga. (2015). Hybrid Association Rule Learning and Process Mining for Fraud Detection. *IAENG International Journal of Computer Science*, p1-14.

[11] Aaron Nech Ira Kemelmacher-Shlizerman. (2017). Level Playing Field for Million Scale Face Recognition. *IEEE*, p1-10.

[12] Nauman Shahid,Ijaz Haider Naqvi,Saad Qaisar,. (2015). One-class support vector machines Analysis of outlier detection for wireless sensor networks in harsh environments. *IEEE*, p1-50.

[13] Nauman Shahid,Ijaz Haider Naqvi,Saad Qaisar,. (2015). One-class support vector machines Analysis of outlier detection for wireless sensor networks in harsh environments. *IEEE*, p1-50.

[14] Erich Schubert, Alexander Koos, Tobias Emrich, Andreas Zufle, Klaus Arthur Schmid, Arthur Zimek. (2015). A Framework for Clustering Uncertain Data. *Proceedings of the VLDB Endowmen*. 8 (12), p1-4.

[15] Anna L. Buczak, Member, IEEE, and Erhan Guven. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*. 1 (2), p1-24.

[16] XiaodongJia,ChaoJin,MattBuzza,Jay Lee. (2017). Wind turbine performance degradation assessment based on a novel similarity metric for machine performance curves. *IEEE*, p1-20.

[17] Yongjoo Park, Michael Cafarella, BarzanMozafari. (2017). Visualization Aware Sampling for Very Large Databases. *IEEE International Conference on Data Engineering*, p1-14.

[18] Fadel M. Megahed and L. Allison Jones-Farmer. (2015). A Statistical Process Monitoring Perspective on Big Data. *Elsevier*, p1-21.

[19] Nurjahan Begum Liudmila Ulanova Jun Wang Eamonn Keogh. (2015). Accelerating Dynamic Time Warping Clustering with a Novel Admissible Pruning Strategy. *Elsevier*, p1-10.

[20] SudiptoGuha,NinaMishra,GouravRoy,OkkeSchrijvers. (2016). Robust Random Cut Forest Based Anomaly Detection On Streams. *International Conference on Machine Learning*. 48, p1-10.

[21] YongruiQina , Quan Z. Shenga , Nickolas J.G. Falknera , SchahramDustdarb , Hua Wangc , Athanasios V. Vasilakosd. (2016). When Things Matter: A Survey on Data-Centric Internet of Things. *Preprint submitted to Journal of Network and Computer Applications*, p1-20.

[22] Theodoros Rekatsinas , Xu Chu , Ihab F. Ilyas , Christopher Ré. (2017). HoloClean Holistic Data Repairs with Probabilistic Inference. *IEEE*, p1-13.

[23] Saeed Aghabozorgi, Ali SeyedShirkhorshidin ,Teh Ying Wah. (2015). Time series clustering A decade review. *Information Systems*, p1-23.

[24] Yan Xia Xudong Cao Fang Wen Gang Hua Jian Sun. (2015). Learning Discriminative Reconstructions for Unsupervised Outlier Removal. *IEEE*, p1-8.

[25] MENG JIANG and PENG CUI,ALEX BEUTEL and CHRISTOS FALOUTSOS, SHIQIANG YANG. (2016). Catching Synchronized Behaviors in Large Networks A Graph Mining Approach. *ACM Transactions on Knowledge Discovery from Dat*. 10 (4), p1-27.

[26] McMahon GT, Gomes HE, Hohne SH, Hu TM, Levine BA & Conlin PR (2005), Web-based care management in patients with poorly controlled diabetes. *Diabetes Care* 28, 1624–1629.

[27] Thakurdesai PA, Kole PL & Pareek RP (2004), Evaluation of the quality and contents of diabetes mellitus patient education on Internet. *Patient Education and Counseling* 53, 309–313.