D3O: A Framework for Distributed Distance-based Detection of Outliers in Large Data Sets

[1]K. Ashesh, [2]Dr.G. AppaRao,

[1]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

[2]Professor, Department of Computer Science and Engineering, Gandhi Institute of Technology and Management (GITAM), Visakhapatnam, AP, India

imasheshk@gmail.com; a.gidutui@gitam.edu

Abstract: Data comes from diversified sources in a distributed computing environment. Outlier detection in such environment is challenging as it involves a strategy to mine outliers. Parallel processing of data available in multiple sources can provide outliers in short span of time. In fact speed with which outlier are mined and interpreted to make well informed decisions is very important in many real world applications like disease outburst detection in healthcare domain. Towards this end, in this paper, we proposed a framework known as Distributed Distance-based Detection of Outliers (D30). The framework guides the process of discovering outliers from large data sets. An algorithm named Distributed Outlier Detection (DOD) is proposed to achieve this. The algorithm exploits the notion of outlier detection solving set to have effective detection of outliers. Two synthetic datasets known as G2d and G3d and a real dataset from NASA named 2Mass are used to evaluate the proposed algorithm. We built a prototype application to demonstrate proof of the concept. The empirical results revealed that the proposed algorithm is capable of finding outliers effectively. The algorithm showed better performance when compared with other state of the art outlier detection algorithm that employs distributed approach in mining outliers.

Index Terms – Outliers, distance-based outlier detection, distribute detection of outliers

1. INTRODUCTION

Outlier is an instance of object that is peculiar when compared with the other objects or instances. Outlier detection is one of the areas in data mining domain which has plenty of applications in the real world such as image processing, anomaly detection, intrusion detection, data cleaning, fraud detection and disease outbreak detection to mention few. When no training data is available, outlier detection can be done using unsupervised learning approach. Machine learning algorithms in this category can discover outliers by grouping exceptional objects and normal objects. Machine learning algorithms related to supervised learning on the other hand need training data in order to learn a model for detecting outliers. Then the classifier can be used to work on the testing data. Many approaches found in the literature, as described below, attempt to discover outliers from large datasets. However, they are different in a measurement or weight or score that is used to determine outliers or distinguish them from normal instances. However, in the real-world data comes from diversified sources and there is often need for distributed processing. There is need for multiple servers to get involved in the processing to have quick discovery of outliers from large databases.

Many researchers focused on the problem of outlier detection as explored in [5], [6], [10], [11], [14]-[16]. Improved genetic k-means is used in [5] for outlier detection. The concept of neighbourhood rank difference is used as a measure to identify outlier detection in [14]. Outlier detection using data mining in Wireless Sensor Network (WSN) is studied in [15] while a hybrid approach that combines weighted k-means and neural networks in order to reduce number of outliers. Most of the existing algorithms are designed to work in a single processor. They are not distributed in nature. The work close to this paper is in [21] where outlier detection is made in distributed approach. In this paper we proposed a framework for distributed distance-based detection of outliers. Our contributions are as follows.

We proposed a framework known as Distributed Distance-based Detection of Outliers (D30) for extracting hidden outliers from large data sets in distributed environment. The framework provides mechanisms to handle data in such environment and produce outliers that are extracted faster besides providing the utility to make well informed decisions.

We proposed an algorithm named Distributed Outlier Detection (DOD) which considers the notion of outlier detection solving set to for better prediction of latent outliers. It works faster besides preserving accuracy.

A prototype application is built to have intuitive user interface to work with datasets from diversified sources. We used two synthetic datasets known as G2d and G3d and a real dataset from NASA named 2Mass. The algorithm is evaluated and compared with other state-of the-art algorithm and found to be performing better.

The remainder of the paper is structured as follows. Section 2 provides review of literature on various outlier detection methods including approaches that run in distributed environment. Section 3 formulates the problem statement. Section 4 presents preliminaries to understand the proposed framework. Section 5 presents the proposed D30 framework. Section 6 presents experimental results. Section 7 concludes the paper besides providing directions for future work.

2. RELATED WORK

This section provides review of literature on different outlier detection methods. Agrawal et al. [1] made a survey of anomaly detection approaches based on data mining. Their methodology includes parameterization, training, model building and detection. Similar kind of work is made on temporal data in [2]. Spatio-temporal data streams are considered for detection of outliers. Capozzoli et al. [3] on the other hand explored anomaly detection methods using data mining techniques on geographical datasets. Fault detection in smart buildings was the aim of their method. Similar kind of work for fault diagnosis is made in [9]. Miller et al. [4] employed an outlier detection method to identify spam in Twitter data streams. They employed many streaming algorithms in order to achieve this. Harghny and Taloba [5] proposed an improved Genetic K-Means algorithm for detection of outliers with high accuracy.

Schubert et al. [6] proposed a generalized outlier detection method. They employed flexible kernel density estimate functions to achieve this. They used domain knowledge integrated with specific needs of the application to achieve this. An anomaly detection approach is built in [7] using data mining techniques. They evaluated their work with NSL-KDD dataset. Rawte and Srinivas [8] proposed a methodology for fraud detection in case of healthcare domain using data mining techniques. Liu et al.

[10] proposed a methodology for outlier detection using imperfect data labels. It employed LOF-based and SVDD-based learning approaches to achieve this.

Jabez and Kuthukumar [11] proposed outlier-based Intrusion Detection System (IDS) to protect computing network from attacks. They employed a measure known as Neighbourhood Outlier Factor (NOF) to validate their work. Guo et al. [12] on the other hand proposed an outlier detection approach on traffic flows. They employed a method known as short-term traffic conditional variance prediction. They found the relation between a forecasting prediction system and outliers. Wijayasekara et al. [13] proposed a fuzzy logic for anomaly detection in case of building emergency management system. The concept of neighbourhood rank difference (NRD) is used by Bhattacharya et al. [14] for outlier detection. They proposed a score for measuring outlierness of instances identified. Govindarajan and Abinaya [15] explored an outlier detection approach in Wireless Sensor Network (WSN) using data mining techniques.

Lekhi and Mahajan [16] employed hybrid approaches in data mining to reduce outliers. They used K-means and neural network methods to have a hybrid approach for outlier detection. Hayes et al. [17] used big data collected from sensors in order to find contextual anomalies as part of predictive modelling. A dynamic KDA model is proposed by Vadoodparsat et al. [19] for detecting fraudulent transactions. Albashrawi [20] employed data mining techniques to detect fraud cases in financial transactions. In the literature it is found that data miningapproaches and methods are used for outlier detection. However, there is little information found on the outlier detection in the distributed environment. In this paper we proposed a distributed approach to detect outliers in large datasets. It will be useful in many real time applications such as disease outbreak prediction that demand accuracy and lease response time.

3. PROBLEM FORMULATION

Outlier is an abnormal entity or instance in a given dataset. There are many data mining algorithms that discover patterns or trends that are latent from datasets in the real world. However, those algorithms target to obtain business intelligence. It is understood that outliers also provide required business intelligence in many applications like fraud detection in banking or financial transactions, finding disease outbreak and so on. Especially when data comes from diversified sources and that needs to be processed to know the anomalies, it is important to have mechanisms to derive business intelligence. The motivating scenario in this regard is disease outbreak detection in healthcare domain where faster processing of data coming from distributed environment has to be made to arrive at the unbiased conclusions and make strategies to overcome the situation. Towards this kind of solution, in this paper we proposed a framework known as Distributed Distance-based Detection of Outliers (D3O).

4. OVERVIEW OF THE PROPOSED D30 FRAMEWORK

The proposed framework is known as Distributed Distance-based Detection of Outliers (D3O). It is distributed in nature as multiple servers are involved in processing. It is distance-based outlier approach though. The main concept employed in the framework is outlier detection solving set (ODSS). ODSS is nothing but is a subset denoted as S of a complete data set denoted as D. The S has required number of objects taken from D in order to become representative of the D. The distances between the pairs of S x D can help in extracting latent and top-k outliers. There are many considerations like the size of solving set, the number of server nodes involved in the computations and time required to compute pair wise distances among objects.

A. The Framework

The framework proposed in this paper is shown in Figure 1. It is in a distributed environment where there are number of nodes involved in the processing. Each node has specific job to be carried out. The nodes are of two types. They are known as supervisor nodes and worker nodes. Often one supervisor node is sufficient though it is possible to have many. The supervisor node is responsible to delete two things to worker nodes. The delegation of computing process to obtain the outliers expected and the synchronization of results obtained after completion of given task is the two important activities of the supervisor node. On the other hand the worker nodes have many responsibilities.

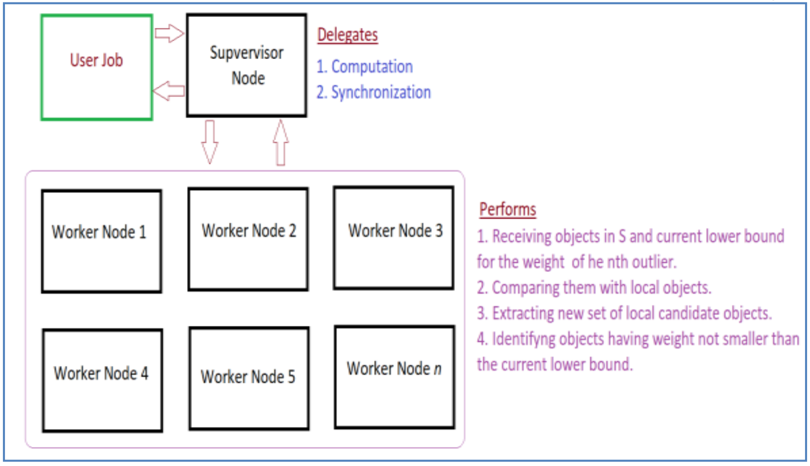
User Job
Supvervisor Node
Delegates
1. Computation
2. Synchronization
Worker Node 1
Worker Node 2
Worker Node 3
Worker Node 4
Worker Node 5
Worker Node n
Performs
1. Receiving objects in S and current lower bound for the weight of he nth outlier.
2. Comparing them with local objects.
3. Extracting new set of local candidate objects.
4. Identifyng objects having weight not smaller than the current lower bound.

Figure 1: Overview of the D3O framework

The worker nodes receive the current solving set objects from the supervisor node along with the current lower bound for nth outlier's weight. Then the worker node is expected to compare them with the local objects. Afterwards, worker node extracts new candidate set (local objects). It is nothing but the objects that have higher weights. Then the weight of local objects is compared with that of current lower bound. Based on this comparison, the worker node determines whether a given object is an outlier or not. Redundant comparisons are carefully avoided to optimize the proposed algorithm. Once synchronization is made, the supervisor node makes new set of global candidates that can be used in the ensuing iteration.

B. Distributed Outlier Detection (DOD) Algorithm

This algorithm is meant for producing outliers with distance based distributed outlier detection strategy in distributed environment. Table 1 shows various notations used in the paper. The algorithm takes dataset as input and computes a solving set in order to produce top-k outliers.

Table 1: Notations used

Table	Notations
act_i, act	Number of global, active objects
C_i,C	Global and local set of candidates

d_i,d	Size of global and local data set
DSS	Distributed solving set
get_k_NNC	This function returns k smallest distance those received in input; it is employed to compute the true k nearest neighbors of the candidate objects.
K	Number of objects for weight calculation
L	Number of local nodes
LC_i	Local candidates
LNNC_i	Local nearest neighbors for candidates
M	Number of objects to be added to the solving set at each iteration
minOUT	Lower bound to the weights of the top n-outliers
N	Number of outliers
NNC	Distances to nearest neighbors for candidates
OUT	Outliers

As shown in Table 1, it is evident that the notations used in the paper are provided with their description. The proposed distributed outlier detection algorithm is as follows.

```
DSS =ϕ
OUT= ϕ
d=$\sum_{i=1}^{l} d_i$
for each node $N_i \in N$
    NodeInit([m$\frac{d_i}{d}$],$c_i$)
C=$\bigcup_{i=1}^{l} C_i$
Act=$d_i$
minOUT=0;
while(C≠ ϕ){
    DSS=DSS ∪C;
    for each node $N_i \in N$
        Nodecomp(minOUT,C,act,LNN$C_i$, L$C_i$,$act_i$;
    act =$\sum_{i=1}^{l} act_i$
    for each q∈ $C${
        NNC[q]=get_K_NNC($\bigcup_{i=1}^{l}$ LNN$C_i[q]$]);
        updateMax(OUT,{q,sum(NNC[q])})
    }
    minOUT=Min (OUT);
    C= ϕ;
    For each p ∈ $\bigcup_{i=1}^{l}$ L$C_i$
        C=C∪ $\{P\}$;
}
```

Algorihm1: Distributed outlier detection algorithm

The algorithm makes use of many inputs such as number of local nodes, size of local datasets, distance function associated with the objects in the dataset, number of neighbours in order to compute weight, the number of top outliers to be found, and an integer telling the number of objects to be added to solving set in each iteration of the process. The algorithm produces solving set and finally top k outliers.

C. Dataset Description

Datasets such as G3d, G2d and 2Mass as explored in [21] are used for experiments. G3d is a synthetic dataset which contains 3D real vectors. It has 500,000 instances. The G2d is also a synthetic dataset containing 1,000,000 instances. The 2Mass dataset contains data of NASA. It has 1,623,376 instances. Table 2 provides summary of datasets used for experiments.

Table 2: Details of datasets

Dataset Name	Number of Instances	Type of Dataset
G3d	500,000	Synthetic
G2d	1,000,000	Synthetic
2Mass	1,623,376	From NASA

The results of experiments made with these datasets are presented in Section 6. The datasets are chosen carefully to be useful for the task in hand. The distributed approach as presented in Figure 1 is used to exploit the dataset and evaluate the proposed algorithm.

5. EXPERIMENTAL RESULTS

Experiments are made with the prototype application using the three datasets described. The observations made include speedup, communication, total execution time and the execution time of supervisor node.

Table 3: Speedup values with different datasets against number of nodes

No of Nodes	**Speedup Values**					
	G3d (Existing)	**G3d (Proposed)**	**G2d (Existing)**	**G2d (Proposed)**	**2Mass (Existing)**	**2Mass (Proposed)**
0	1	4.5	7	11	12	13
5	1	5	8	12	13	14
10	1	5	9	13	14	15
15	1	5.5	9.5	13.5	15	16
20	1	5	10	14	16	17
25	1	5.5	11	14.5	17	18

As shown in Table 2, it is evident that the number of nodes and the speed up values for three datasets in existing and proposed systems are presented.

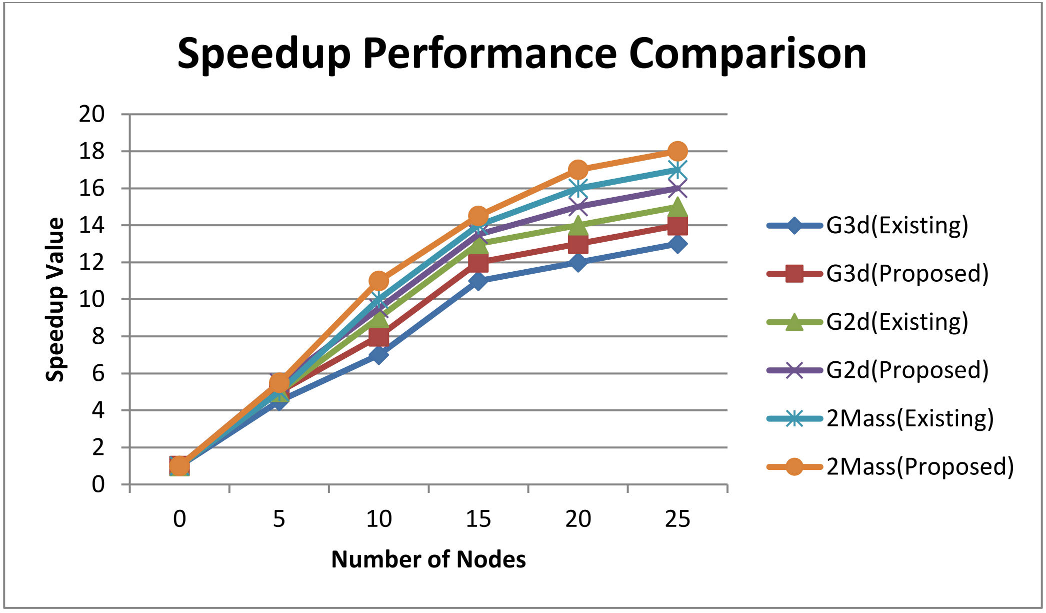

Figure 2: Speedup performance comparison

As presented in Figure 2, it is evident that the horizontal axis shows number of nodes while the vertical axis provides speedup values. The results revealed that the number of nodes has its impact on the speed up value. The proposed system showed better speed up values for all the datasets consistently.

Table 4: Ratio between total execution time and communication time with different datasets against number of nodes

No of Nodes	Ratio between Total Execution Time and Communication Time					
	G3d (Existing)	G3d (Proposed)	G2d (Existing)	G2d (Proposed)	2Mass (Existing)	2Mass (Proposed)
0	0	0.01	0.02	0.06	0.1	0.12
50	0	0.015	0.02	0.07	0.08	0.14
10	0	0.01	0.015	0.04	0.06	0.1
15	0	0.015	0.018	0.06	0.08	0.12
20	0	0.008	0.014	0.03	0.05	0.09
25	0	0.009	0.016	0.04	0.06	0.1

As shown in Table 3, it is evident that the number of nodes and the ratio between total execution time and communication time for three datasets in existing and proposed systems are presented.

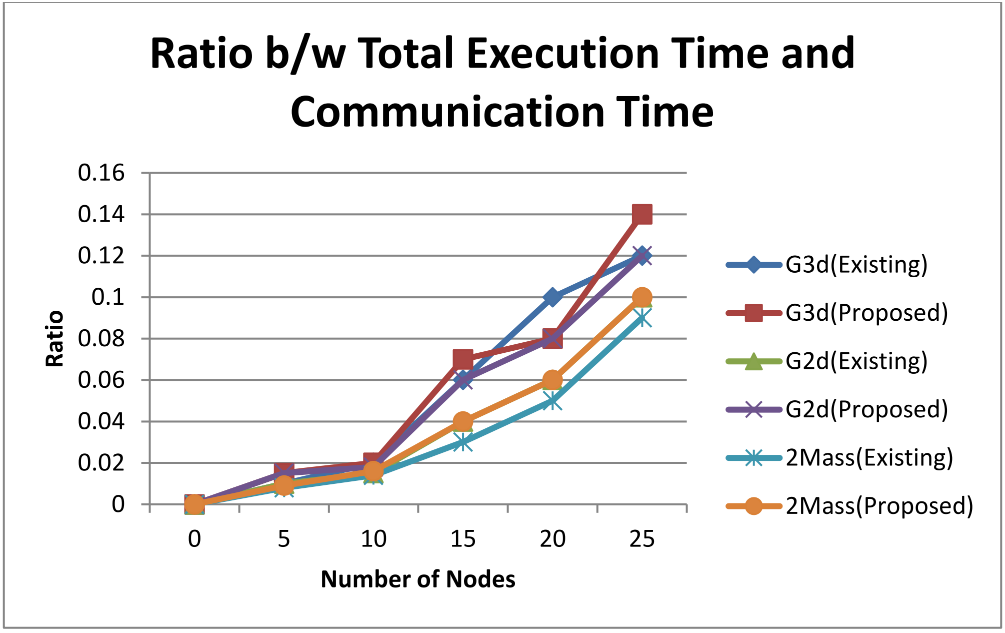

Figure 3: Ratio between Total execution time and communication time performance comparison

As presented in Figure 3, it is evident that the horizontal axis shows number of nodes while the vertical axis provides ratio. The results revealed that the number of nodes has its impact on the ratio between Total execution time and communication time value. The proposed system showed better ratio between Total execution time and communication time for all the datasets consistently.

Table 5: Ratio between total execution time and supervisor node time with different datasets against number of nodes

No of Nodes	**Ratio between the Total Execution Time and Supervisor Node Time**					
	G3d (Existing)	**G3d (Proposed)**	**G2d (Existing)**	**G2d (Proposed)**	**2Mass (Existing)**	**2Mass (Proposed)**
0	0	0.003	0.005	0.01	0.02	0.03
50	0	0.004	0.006	0.012	0.022	0.04
10	0	0.001	0.004	0.008	0.01	0.012
15	0	0.002	0.005	0.009	0.012	0.014
20	0	0.005	0.01	0.012	0.014	0.016
25	0	0.006	0.012	0.014	0.016	0.018

As shown in Table 4, it is evident that the number of nodes and the ratio between total execution time and supervisor node time for three datasets in existing and proposed systems are presented.

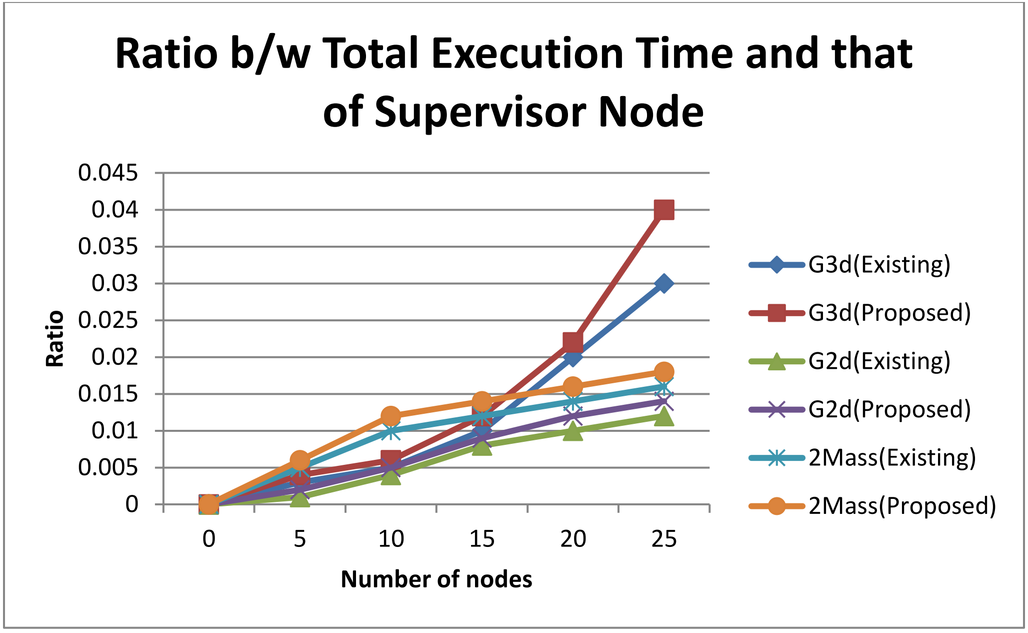

Figure 4: Ratio between Total execution time and that of Supervisor node performance comparison

As presented in Figure 4, it is evident that the horizontal axis shows number of nodes while the vertical axis provides ratio between Total execution time and that of Supervisor node. The results revealed that the number of nodes has its impact on the ratio between Total execution time and that of Supervisor node. The proposed system showed better ratio between Total execution time and that of Supervisor node for all the datasets consistently.

Table 6: No. of relevant equivalent distances with different datasets against number of nodes

No of Nodes	**Number of Relevant Equivalent Distances**					
	G3d (Existing)	**G3d (Proposed)**	**G2d (Existing)**	**G2d (Proposed)**	**2Mass (Existing)**	**2Mass (Proposed)**
0	2	3	2	2	2	2
50	2.5	3.5	2.5	2.5	2.5	2.5
10	2	3	2	2	2	2
15	2.5	3.5	2.5	2.5	2.5	2.5
20	1	2	1	1	1	1
25	1.5	2.5	2	2	2	2

As shown in Table 5, it is evident that the number of nodes and number of relevant equivalent nodes for three datasets in existing and proposed systems are presented.

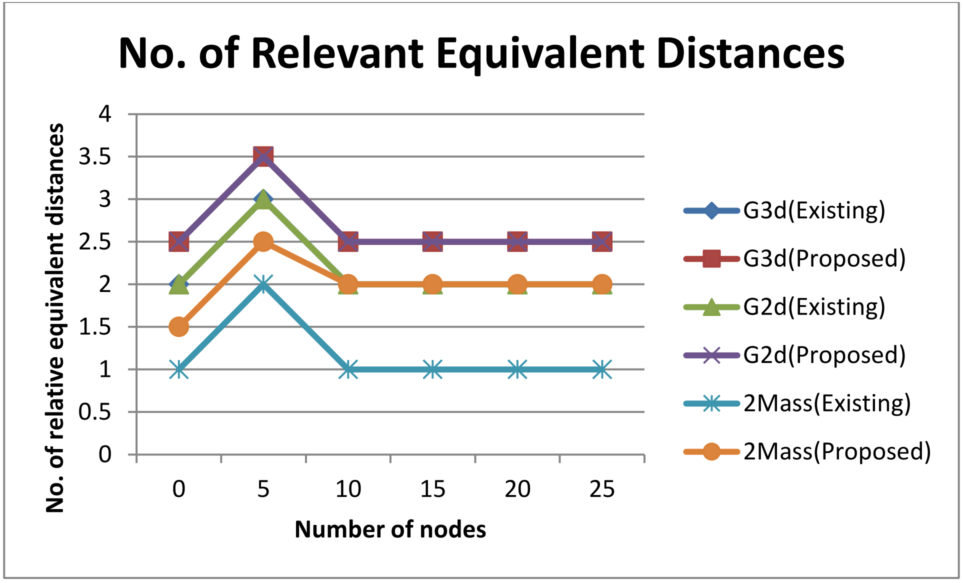

Figure 5: No of Relevant equivalent distances performance comparison

As presented in Figure 5, it is evident that the horizontal axis shows number of nodes while the vertical axis provides No of Relevant equivalent distances the results revealed that the number of nodes has its impact on the No of Relevant equivalent distances. The proposed system showed better No of Relevant equivalent distances for all the datasets consistently.

Table 7: No. of iterations with different datasets against number of nodes

No of Nodes	No. of Iterations					
	G3d (Existing)	G3d (Proposed)	G2d (Existing)	G2d (Proposed)	2Mass (Existing)	2Mass (Proposed)
0	40	40	40	40	40	40
50	45	45	45	45	45	45
10	30	30	30	30	30	30
15	35	35	35	35	35	35
20	20	20	20	20	20	20
25	25	25	25	25	25	25

As shown in Table 6, it is evident that the number of nodes and number of iterations for three datasets in existing and proposed systems are presented.

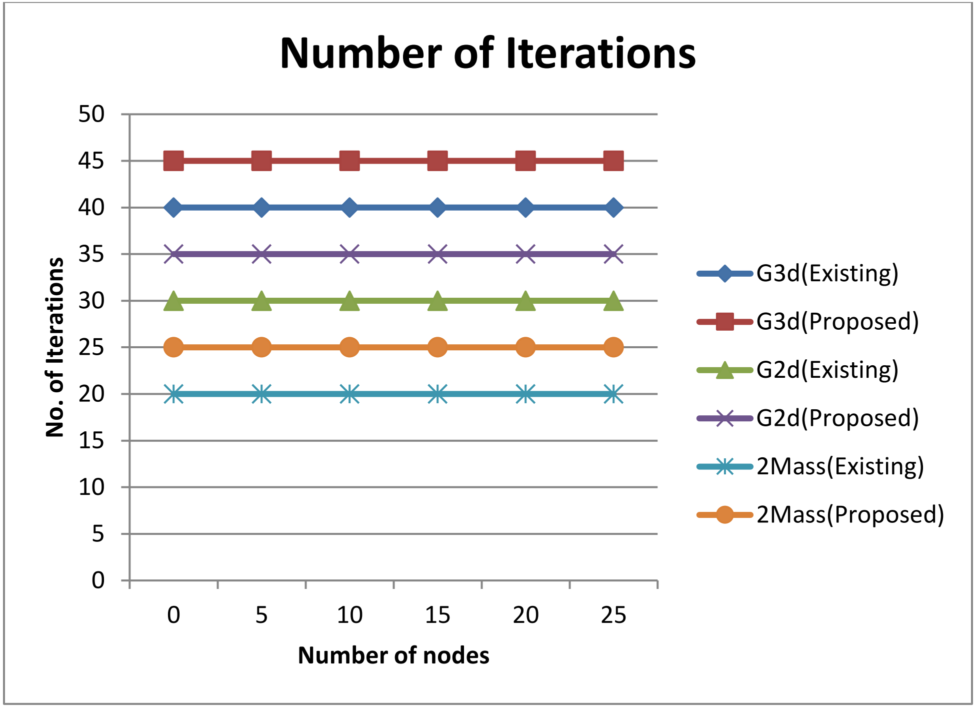

Figure 6: No of iterations performance comparison

As presented in Figure 6, it is evident that the horizontal axis shows number of nodes while the vertical axis provides no of iterations performance .The results revealed that the number of nodes has its impact on No of iterations performance. The proposed system showed better no of iterations performance for all the datasets consistently.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a framework known as Distributed Distance-based Detection of Outliers (D30) to discover outliers from large datasets. It makes use of distributed approach in dealing with diversified sources of data. An algorithm by name Distributed Outlier Detection (DOD) is proposed to achieve this. The algorithm employs the notion of outlier detection solving set in tune with the distributed approach. Two synthetic datasets known as G2d and G3d and a real dataset from NASA named 2Mass are used to evaluate the proposed algorithm. The performance metrics used to evaluate the algorithm include speedup, ratio between the total execution time and communication time and the ratio between the total execution time and the supervisor node time. We built a prototype application to demonstrate proof of the concept. The experimental results revealed that the proposed framework shows improved performance over a state of the art algorithm found in the literate with all the datasets considered. An important direction for future work is to explore data mining algorithms like Support Vector Machines (SVM) for mining outliers in distributed environments. Another direction is to have an ensemble method for distributed outlier detection to improve accuracy and performance in detection of outliers.

7. REFERENCES

1. Shikha Agrawal AND Jitendra Agrawal. (2015). Survey on Anomaly Detection using Data Mining Techniques. elsever, P708 – 713.
2. Manish Gupta, Jing Gao, Charu C. Aggarwal and Jiawei Han. (2014). Outlier Detection for Temporal Data, A Survey. IEEE. 26 (9), P2250-2267.
3. Alfonso Capozzoli, Fiorella Lauro AND Imran Khan. (2015). Fault detection analysis using data mining techniques for a cluster of smart office buildings. elsever, P4324–4338.
4. Zachary Miller, Brian Dickinson, William Deitrick, Wei Hu AND Alex Hai Wang. (2012). Twitter Spammer Detection Using Data Stream Clustering, p1-19.
5. M. H. Marghny AND Ahmed I. Taloba. (2011). Outlier Detection using Improved Genetic K-means. International Journal of Computer Applications. 28 (11), p1-4.
6. Erich Schubert, Arthur Zimek AND Hans-Peter Kriegel. (2014). Generalized Outlier Detection with Flexible Kernel Density Estimates. International Conference on Data Mining, p1-9.
7. SolaneDuquea, Dr.Mohd AND Nizam bin Omarb. (2015). Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS). elsever, P46 – 51.
8. VipulaRawte AND G Anuradha. (2015). Fraud Detection in Health Insurance using Data Mining Techniques. International Conference on Communication, Information & Computing Technology, P1-6 .
9. AfroozPurarjomandlangrudi, Amir Hossein Ghapanchi AND Mohammad Esmalifalak. (2014). A data mining approach for fault diagnosis, An application of anomaly detection algorithm. elsever, P343–352.
10. Bo Liu, Yanshan Xiao, Philip S. Yu, Zhifeng Hao, and Longbing Cao. (2014). An Efficient Approach for Outlier Detection with Imperfect Data Labels. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. 26 (7), P1602-1616.
11. JABEZ Ja AND Dr.B.MUTHUKUMAR. (2015). Intrusion Detection System (IDS), Anomaly Detection using Outlier Detection Approach. elsever, P338 – 346.
12. Jianhua Guo, Wei Huang AND Billy M. Williams. (2014). Real time traffic flow outlier detection using short-term traffic conditional variance prediction. elsever, P1-13.
13. DumiduWijayasekara, Ondrej Linda, Milos Manic and Craig Rieger. (2014). Mining Building Energy Management System Data Using Fuzzy Anomaly Detection and Linguistic Descriptions. IEEE, p1-12 .
14. Gautam Bhattacharyaa, Koushik Ghoshb AND Ananda S.Chowdhury. (2015). Gautam Bhattacharyaa, Koushik Ghoshb, Ananda S.Chowdhury. elsever, P24–31.
15. M.Govindarajan and V.Abinaya. (2014). An Outlier detection approach with data mining in wireless sensor network. International Journal of Current Engineering and Technology, p1-4 .
16. NancyLekhi AND Manish Mahajan. (2015). Outlier Reduction using Hybrid Approach in Data Mining. I.J. Modern Education and Computer Science, P43-49.
17. Michael Hayes AND Miriam A M Capretz. (2013). Contextual Anomaly Detection in Big Sensor Data. Electrical and Computer Engineering, p1-9.
18. M.Vadoodparast, Prof. A. RazakHamdan AND Dr. Hafiz. (2015). FRAUDULENT ELECTRONIC TRANSACTION DETECTION USING DYNAMIC KDA MODEL. International Journal of Computer Science and Information Security. 13 (2), P1-10.
19. Mousa Albashrawi. (2016). Detecting Financial Fraud Using Data Mining Techniques, A Decade Review from 2004 to 2015. Journal of Data Science, P553-570.
20. JABEZ Ja AND Dr.B.MUTHUKUMAR. (2015). Intrusion Detection System (IDS), Anomaly Detection using Outlier Detection Approach. Elsever, P338 – 346.

21. Fabrizio Angiulli, Stefano Basta, Stefano Lodi and Claudio Sartori (2013). Distributed Strategies for Mining Outliers in Large Data
    Sets. IEEE Transactions on Knowledge and Data Engineering, 25(7), p1520-1532.
22. Shaik, Hasane & Misra, Yogesh & Bojja, Polaiah. (2017). A review of application of fuzzy controller in sugar industry. Journal of Advanced Research in Dynamical and Control Systems. 2017. 34-47.
23. Ahammad, S. H., Rajesh, V., Indumathi, U., &Charan, A. S. (2019). Identification of Cervical Spondylosis disease on Spinal Cord MRI Image using Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) Technique. vol, 46, 108-124.
24. Vallabhaneni, R. B., & Rajesh, V. (2018). Brain tumour detection using mean shift clustering and GLCM features with edge adaptive total variation denoising technique. Alexandria engineering journal, 57(4), 2387-2392.
25. Ratna, Bhargavi & Rajesh, V.. (2018). Computer Aided Bright Lesion Classification In Fundus Image Based On Feature Extraction. International Journal of Pattern Recognition and Artificial Intelligence.
26. Vallabhaneni, R. B., & Rajesh, V. (2017). On the performance characteristics of embedded techniques for medical image compression.
27. Ahammad, S. H., Rajesh, V., Neetha, A., Sai Jeesmitha, B., & Srikanth, A. (2019). Automatic segmentation of spinal cord diffusion MR images for disease location finding. Indonesian Journal of Electrical Engineering and Computer Science, 15(3), 1313-1321.
28. Ramesh Babu Vallabhaneni and V. Rajesh, 2017. Performance Analysis of Total Variant Techniques for Efficient Segmentation of Medical Images. Journal of Engineering and Applied Sciences, 12:5343-5346.
29. Gajula, Srinivasarao& Rajesh, V.. (2018). Enhanced medical image watermarking scheme with CLA-HE & DWT, SVD transforms. International Journal of Engineering and Technology (UAE). 7.2603
30. Ahammad, S. H., Rajesh, V., & Rahman, M. Z. U. (2019). Fast and accurate feature extraction-based segmentation framework for spinal cord injury severity classification. IEEE Access, 7, 46092-46103.
31. Dudi, B. & Rajesh, V.. (2018). An efficient algorithm for medicinal plant recognition. International Journal of Pharmaceutical Research. 10. 87-93.
32. Ahammad, S., Rajesh, V., Saikumar, K., Jalakam, S., & Kumar, G. N. S. (2019). Statistical analysis of spinal cord injury severity detection on high dimensional MRI data. International Journal of Electrical & Computer Engineering (2088-8708), 9.

Mr.K.Ashesh is currently a Research Scholar with the Department of Computer Science and Engineering, GITAM institute of Technology, GITAM University, Visakhapatnam, India.

Dr.G.Apparo is currently a Professor with the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM University, Visakhapatnam, India. His Research focuses on Data Mining.