Efficient Clustering Based Local Outlier Detection Over High Dimensional Streaming Data

[1] Ms.Ashwini Jadhav, [2]Prof . K . V. Metre

[1, 2]Department of Computer Engineering,
[1, 2]MET'S BKC IOE ,Nashik, Maharashtra

***Abstract :** : Local and global outliers detection are two important aspects in distance based outlier detection process. To analyze outliers from high speed data streams is challenging task. In existing outlier detection techniques, along with the incoming data stream points, previous data points are also stored for processing with the assumption of unbounded memory. To overcome this problem, memory efficient incremental outlier detection over data stream-(MiLOF) is also proposed in existing work. This technique preserves the summary of previous data points and finds local outliers. In this research work Memory efficient local outlier detection technique is proposed over data stream with feature extraction process. This technique preserves the summary of previous data points in each iteration and finds local outliers. This algorithm is divided is 3 phases namely: Summarization, Merging and Revised Insertion. To reach the memory constraint criteria summarization is used. To generate revised clusters merging phase is used. Whereas in revised insertion, LOF is identified and points are placed in cluster or marked as an outlier. Feature extraction process helps to reduce dimension in high dimensional dataset and hence to reduce memory for processing.*

***IndexTerms** - outlier detection, local outlier, clustering, density based outlier, data stream, feature extraction.*

I. Introduction

In knowledge discovery process, mining of useful data is done. But there is less research work is done in finding exceptions in data. Outlier is the unexpected behaviour of data point. Outlier detection is used to find rare events, exceptional cases or some sort of deviation from regular entries. This is applicable in various domains such as: detecting criminal activities in bank transitions or digital market, intrusion detection, etc.

The outlier detection strategy varies with respect to the given input data. For example outlier detection in credit card transaction is different from outlier in meteorological data. Outlier is unexpected behaviour of data point but this is much generalized approach to define the outlier. The definition and outlier detection treatment varies with respect to the application. Various approaches are used for outlier detection such as supervised, semi-supervised, unsupervised. In supervised approach a labelled data is provided as an training data to the system. The labels include entries of inliers as well as entries of outlier. Based on the training dataset, outliers in test datasets are identified. In semi-supervised approach only inliers or outliers are labelled as a training data. In unsupervised approach no labelled data is provided to the system for training. The unsupervised approach is widely used in variety of domains because of unavailability of labelled data.

The unsupervised outlier detection technique is mainly classified in 3 categories:

1. **Distribution based:** In this technique outlier is identified using probabilistic distribution of data.
2. **Depth Based:** In this technique each point in dataset is treated as k-d space called as depth. Outliers are those points whose k-depth is minimum.
3. **Distance based:** It uses k –nearest neighbour technique to rank the outlier from a given dataset.

In variety of applications streaming data is generated. Streaming data is continuous unbounded sequence of data records. It can be ordered by explicit timestamp. The stream data processing includes variety of aspects such as system design, resource optimization, scalability storage management, etc. But very less attention is provided on outlier detection over streaming data.

The outlier detection over streaming data is difficult task because volume of data is generated continuously. To analyze such unbounded data is challenging task. The whole data can not be store in memory for processing. Some resources where streaming data is captured are configured with limited memory such as wireless sensor networks. Such devices need memory efficient outlier detection strategy.

The following section includes the literature work related to the static data outlier detection strategies and streaming data outlier detection strategies. This strategy follows the unsupervised approach for outlier detection.

II. Related Work:

Outlier and anomaly are two terms that can be interchangeably used for outlier detection. A review related to the anomaly detection techniques are provided by author . In this review anomaly is categorized in 3 types: Point Anomalies, Contextual Anomalies and Collective Anomalies. According to this review anomaly identification strategies are have different implementations as per the identification of anomaly type. It is required to identify which kind of technique will be suitable for the given problem. Most of the existing work focuses on static dataset processing for outlier detection technique. In such cases all the data points are available for processing at a time. This requires high processing time and memory usage. [3]

Wireless sensor networks are used in variety of applications. Anomaly detection on wireless sensor network is challenging task due to memory limitations. Author provides a review on techniques used for outlier detection in wireless sensor network. The solution is problem specific and hardware specific. To provide solution over sensor network multiple aspects should be considered such as: communication frequency, minimizing energy consumption, etc .[4]

Streaming data processing is proposed and where the processing is categorized in 3 sections. : Distribution based, clustering based and distance based.[5][6][7]

Distribution based technique aims to learn probabilistic distribution of data. This requires a-priory knowledge of distribution of data which is impractical in streaming data solutions. [8]

In cluster based approach more focus is on cluster creation than outlier detection. In this technique, outliers are detected those are far away from the centroid based on small clusters or data points. There is no efficient scheme provided for high dimensional data clustering and outlier detection. n. It proposes a cluster based histogram. This histogram is used to model streaming data in applications. Histogram generates summary of data points. Supervised and unsupervised data solutions are proposed in this work based on the underlying input data structure. It proposed a cluster based outlier detection for streaming data. This technique mainly includes parameter free algorithm. A self adaptive algorithm is proposed to tackle with varying data arrival rate in data stream. [9][10] [11] [12]

Distance based approach calculates the distance of each point with respect to the remaining points. The distance based outlier detection technique is mainly classified in two categories :

A. Global Outlier Detection:

For global outlier detection, the distance of data point is compared with the all the data points present in memory. This is generally applicable in static dataset. [13]

A sliding window technique is applied to find global outliers based on previous input. The outliers in current window are detected by considering the whole dataset. Author proposed an editing based approach to find global outliers over streaming data. This approach is supervised approach and hence applicable in very limited applications. [14][15][16]

A combine cluster based and distance based approach to detect global outlier over data stream is proposed in literature. This technique provide better solution than existing cluster based or distance based solution in terms of computational cost. [17]

B. Local Outlier Detection:

Unlike global dataset, local outlier local outlier is detected based on k nearest neighbor form data slice. For static dataset analysis LOF is proposed. In this , degree of being outlier is calculated for each data point. This provides better solution of non-homogeneous data distribution. The complexity of LOF depends on the number of datapoints and is quadratic .To improve efficiency of LOF , approximation technique is proposed. In this technique LOF factors is not calculated for all the data points. This improves efficiency of system. [18][19]

The previous version of LOF requires to preserves previous data points in memory to find outlier in next iteration. Author proposed a local outlier detection technique based on incremental approach over data streams. This technique calculates the LOF value based on k-nearest neighbors and values of LOF will be updated of KNN if required. But In this technique LOF for all incoming data points need to be calculated and hence this method consumes high memory, and high processing time. [5]

Author proposed a memory efficient technique for local outlier detection. In this technique summary of previous data points is preserved rather than preserving whole dataset. But this technique does not efficiently handle the high dimensional dataset with specific memory bound. [1]

Data cube analysis is a very strong tool used for analysis of multidimensional data. Interesting measure computation for data cubes and relative mining of interesting cube groups over data sets at large scale such as web logs are complex for many important analyses done in the real world. Existing approaches have focused on algebraic measures like SUM that are suitable to parallel computation and can easily take advantage from the recent parallel computing infrastructure like MapReduce. Author conclude that, unlike existing techniques which cannot scale to the 100 million tuple mark for our data sets, MR-Cube successfully and efficiently computes cubes with holistic measures over billion-tuple data sets. [20]

There are different approaches are used in the field of data mining like text mining,patteren mining. For mining the high utility itemsets from large transactional datasets multiple methods are available and have some consequential limitations. Performance of these methods need to be scrutinized under low memory based systems for mining high utility itemsets from transactional datasets as well as to address further measures. The author proposed algorithm combines the High Utility Pattern Mining and Incremental Frequent Pattern Mining. Two algorithms used are Apriori and existing Parallel UP Growth for mining high utility itemsets using transactional databases. [21]

Feature selection involves selecting the most useful features from the given data set and reduces dimensionality.A novel clustering approach is proposed for feature selection from high dimensional data. The formation of clusters drastically reduces the dimensionality and helps in selection of relevant features for the concerned target class. The data pre processing removes the redundant and irrelevant features. The formation of clusters by constructing minimum spanning tree reduces the complexity for the computation of feature selection. [22]

System consist of continuous queries are long running queries. These queries are used to monitor the changes to time varying data. The queries are divided among the sub-queries. The simulated annealing algorithm for deriving subqueries is given. In pull based data dissemination mechanism user explicitly requests data items from the server. Push based data dissemination mechanism maintains state information pertaining to clients and push only those changes that are of interest to a user. Various optimization algorithms can be compared for deriving the most optimal solution by author. [23]

The dynamic data is changing very rapidly. The continuous queries are used to get the updated values of this dynamic data. The continuous query is divided in sub queries and forwarded to particular data aggregator by the central node. By using simulated annealing algorithm, the optimized results can be obtained. Push based data dissemination technique is used for dissemination of the results to the client. In push based technique the user does not need to refresh data time to time. As soon as the data is available at the data aggregator the data is disseminated to the client. [24]

There is a need to provide a technique to reduce processing space and data storage space for continuous input data. The incoming data is multidimensional data. There is need to provide dimensionality reduction technique to reduce storage space. There is a need of such system to provide a solution, for local outlier detection within the given memory bound, for incoming streaming data points. The proposed system provides a solution for local outlier detection within the given memory bound, for incoming streaming high dimensional data points. [25]

In this system author discussed various outlier detection techniques. The outlier detection techniques are mainly classified in 3 categories: distribution based, cluster based and distance based. Distance based outlier detection technique is more suitable technique to handle non-homogeneous data density over streaming data. Global and local outlier detection are 2 main aspects in distance based outlier detection technique. To find local outlier over streaming data various factors need to be focused such as: Incoming data structure, memory bounds, data dimensionality, etc [26]

Author have discussed various similarity measures to find the exact nearest neighbours. They discussed about clustering techniques for content based feature extraction from image. [27]

Author have discussed about dynamic data analysis.Continuous queries are used to monitor the changes to time varying data and to provide useful results. In these queries the query incoherency bound is given with the query. The query is divided into the sub-queries. Network of data aggregator serves user with data having options for combinations of aggregators. The goal is to minimize the number of refreshes. The refresh message is sent from the data aggregator to the client. The time varying data is present in financial information such as stock prices, and currency exchange rate, real time traffic and weather information. The query is optimized and the dissemination of the data updates are disseminated through a network of data aggregators. [28]

III. Problem Formulation:

Local and global outliers detection are two important aspects in distance based outlier detection process. To analyze outliers from high speed data streams is challenging task. In existing work lot of work has been done to find local outlier over streamingdata. But these techniques suffer from insufficient memory problems and hence these systems were able to process limited data.For every incoming input stream, points are collected at time t and its Local Outlier Factor-LOF value is calculated. It is impractical to store all the incoming points and its LOF values in the memory after every input stream. There is a need to provide a technique to reduce processing space and data storage space for continuous input data. The incoming data is multidimensional data. There is need to provide dimensionality reduction technique to reduce storage space. There is a need of such system to provide a solution, for local outlier detection within the given memory bound, for incoming streaming data points.

IV. System Architecture:

Following figure 1 represents the architecture of system.

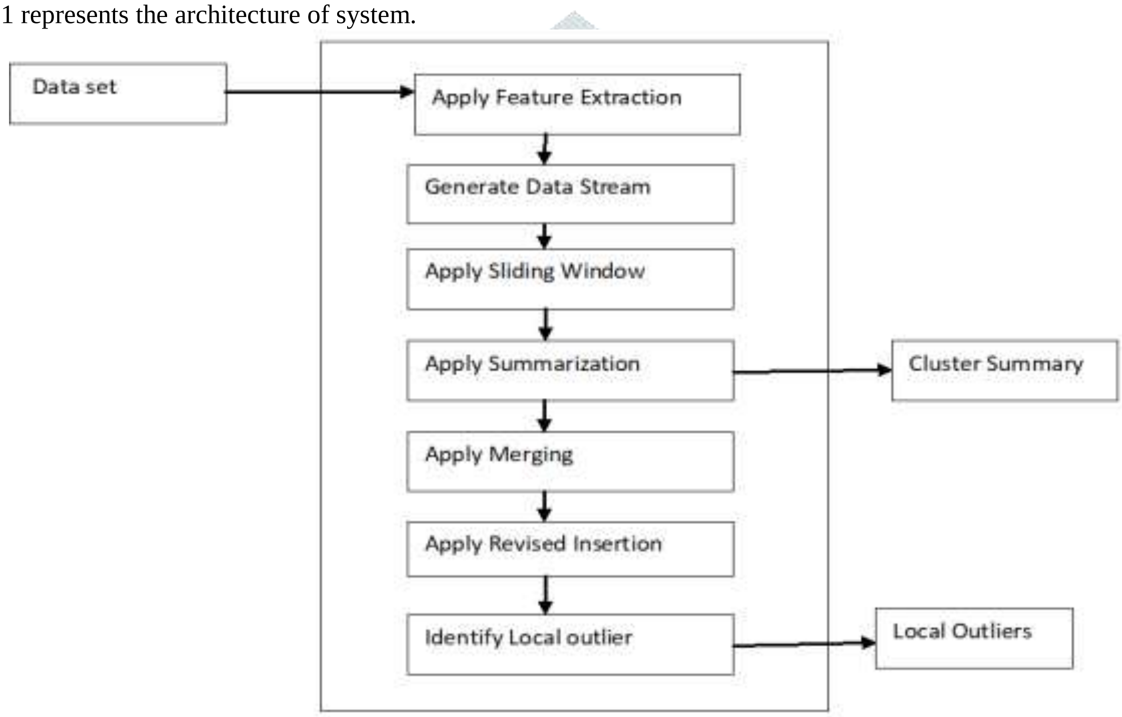


Fig 1.System Architecture

When data arrives for processing initially feature selection technique is applied to reduce the processing dimensional space. Along with the selected feature subset incoming data stream is processed in 4 steps: Attribute Filter, Summarization, Merging and Revised Insertion.

Dimensionality Reduction:

Whole dataset is given to the attribute extraction process. This technique uses Principal Component analysis to select attributes from dataset. This technique merges 2 or more relevant features to the single feature. .

1.Attribute Filter:

The incoming stream data is filtered as per the selected attributes and saved in the memory space.

2.Summarization:

In every data input stream b points are processed. When processing memory reaches to its limit, summary of b/2 data points is calculated and deleted from the memory. The summary is calculated based on k distance ,Local reachability density and LOF values. These values can be calculated as follows:

- **k-distance(p):**

  the distance between a data point p and its kth nearest neighbor (kth-NN).

- **Local reachability density (lrd):**

  Local reachability density (lrd) of a data point p:

$$Lrd_k(p) = \frac{1}{k} * (\Sigma_{o\epsilon N(p,k)}\, Reach - dist(p,o))^{-1}$$

  where N(p,k) is the set of k nearest neighbors of p.

- **Reachability distance:**

  Reachability distance (reach-dist) of a data point p with respect to another data point o:

Reach-dist (p,o) = max(K-distance(o) , d(p,o))

- **Local Outlier Factor LOF:**
  Local outlier factor of a data point p

$$LOF_k(p) = \frac{1}{k} * \Sigma_{o\epsilon N(p,k)} \frac{lrd_k(o)}{lrd_k(p)}$$

3.Merging:
In each iteration, when b/2 points are received clusters are generated for those points with c-means algorithm. These clusters are merged with previous cluster centers generated in previous iteration and stored in summarization phase. The merging is done using weighed clustering algorithm.

4.Revised Insertion:
In revised insertion phase value of LOF is calculated for each data point and outlier is identified. After identification of Outliers, k-distance, reach-dist, lrd and LOF values for the existing data points are updated.

V.Algorithms:

Algorithm 5.1 Flexible C-Means Clustering

```
Input:  data points set Ci
Output:  cluster centers set Vi
         cluster member count set Ni
Processing
 // First clustering of points into c clustering
1.(Vi,{Ci1 U Ci2 U Cic}) By C means
 // clusters with high probability of being outliers are prune
2.λ {k-distance(p) | p Ɛ Ci}
3.POT-Ci μ(Y)+3σ(Y) where Y Ɛ λ
4.For all Cij in Ci do
    if Majority of jth cluster members' k-distance >POT-Ci then
            Remove relevant cluster center from Vi
            end if
        end for
5.Ni {|Ci1|, |Ci2|, |Cic|}
6.Return Vi and Ni
```

Algorithm 5. 2 MiLOF

```
Input:  data points set P = {p1, p2,.., pn},
        memory size limit  m (choose b : c = m - b)
 Output:  LOF set  = {LOF(p1),…,LOF(pn)} values
Processing:
1. Apply dimensionality reduction using PCA
2.for all pt Ɛ P do
3.LOF(pt)=   Revised Insertion(pt)
4. if Number of data points in memory = b then
     load b/2 point p to Ci
5.(Vi;Ni)= c-means(Ci) for Flexible c-means(Ci)
6.for all vij Ɛ V i do
         Compute k-distance(vij, ), lrd(vij), LOF(vij)
7.Delete Ci
      if ctr>0 then
 (Z,W)  = Weighted c-means(Vi U Vi-1, Ni U Ni-1)
8.for all Zj  Ɛ  Z do
      Compute k-distance(Zj ) and lrd(Zj ), LOF(Zj )
      end for
```

```
9. Vi  = Z
      Ni =summation of all W
10.Delete Vi-1, Z
11. Update k-NNs if deleted of remaining points
      return LOF
```

5.3 Datasets:

Datasets are downloaded from UCI repository[20]. Fom this repository Vowel, Letter datasets are downloaded. We have added 5% of noise in original dataset .And we compare our result with vowel dataset .

5.4 Performance Measure:

The performance of a system is depend on parameter b i.e. number of records and number of attributes in dataset. The performance is evaluated in terms of:

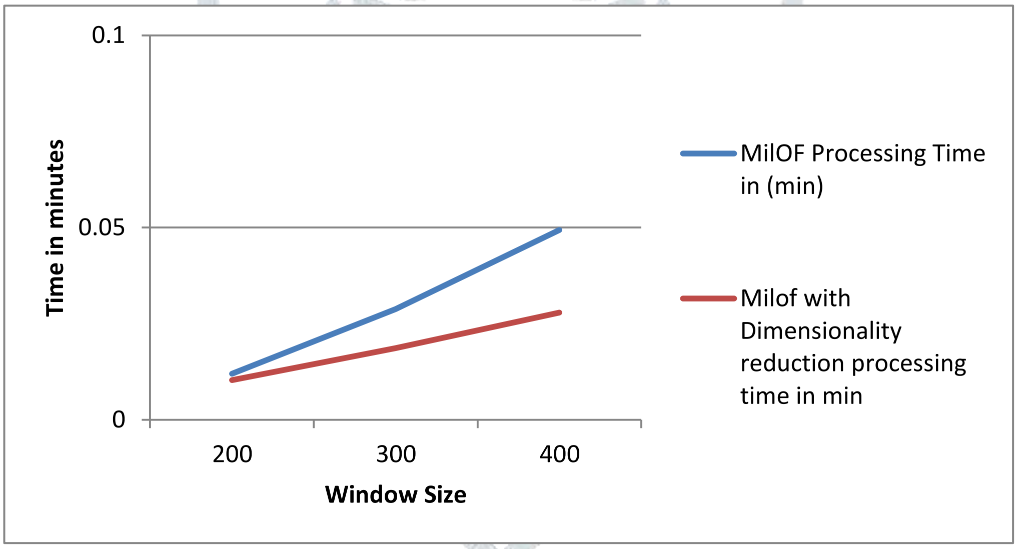
- Time: time required for processing is calculated.
- Memory: The run time memory requirement of the system is calculated.

VI.Results and Discussion:

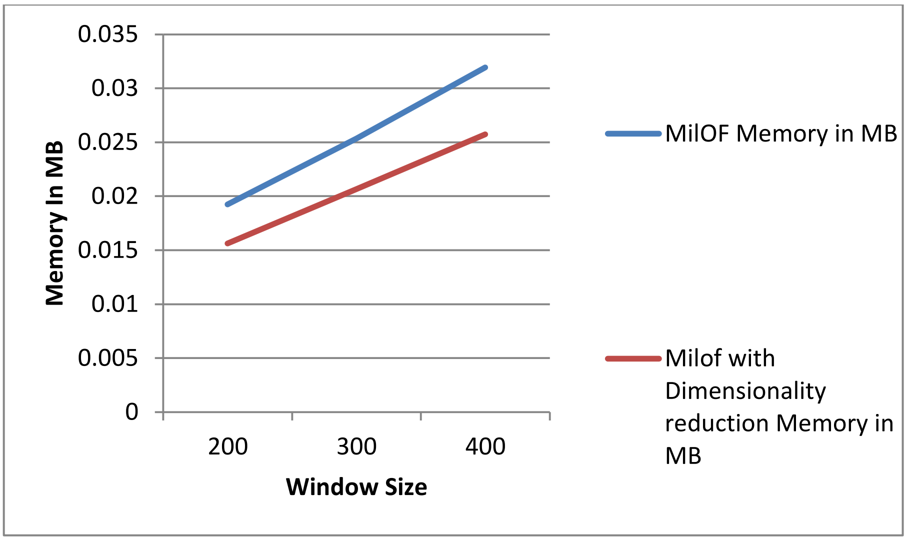
Following table shows the processing time and memory required for processing for MiLOf and MiLOF with dimensionality reduction. The processing time and memory is depend on the window size and number of dimensions in dataset. As we decrease the number of dimensions the processing time and memory required for processing is also reduced.

		MiLOF		MiLOF with Dimensionality Reduction	
Dataset	Window Size	Processing Time(in Min)	Memory (in MB)	Processing Time(in Min)	Memory (in MB)
Vowel	200	0.01200	0.01924	0.0102	0.0156
	300	0.01877	0.02535	0.0186	0.0206
	400	0.02936	0.03193	0.0278	0.0257

Following graph represent the comparison of processing time of systems



Following graph represent the comparison of Memory of systems



Following table shows the number of outlier found based on the various window sizes. As we change the window size the number of outlier found also changes.

Dataset	Window Size	Initial Cluster count	Number Of outlier found using MiLOF with dimensionality reduction
Vowel	200	50	42
	300	50	38
	400	50	26

Following table shows the number of outlier found with various number of cluster count as an input to the system.

Dataset	Window Size	Initial Cluster count	Number Of outlier found using MiLOF with dimensionality reduction
Vowel	200	40	42
	200	50	42
	200	60	42

V. System Comparison:

This table 2 represents the system comparison among existing approach and proposed approach. LOF preserves n points in memory with d dimensions. The MiLOF and D-MiLOF approaches preserves b +c points in memory where b is the number of streaming points and c is cluster summary. The time complexity of MILOF and D-MiLOF is reduced due to reduced data size.

	Number of Points in Memory	Time Complexity	Data Dimension
LOF	N	nLog(n)	D
MiLOF	b+c	n*log(b+c)	D
D-MiLOF(Proposed Ssytem)	b + c	n*log(b+c)	K < d

Conclusion:

A memory efficient local outlier detection solution over steaming data is provided in this system. A MiLOF with dimensionality reduction technique is provided. The system performance and memory requirements are depend on the number of data points for processing and the dimensions in the dataset. The system generates cluster summary from the previous data points and preserve the summary of points in memory. This MiLOF features helps to run the system in limited memory resources. The system performance is checked with various datasets by varying cluster counts, window sizes and attribute counts. In future the system can be implemented on wireless sensor networks for outlier detection.

References:

1.Mahsa Salehi , Christopher Leckie , James C. Bezdek , Tharshan Vaithianathan and Xuyun Zhang, ""Fast Memory Efficient Local Outlier Detection in Data Streams," EEE Transactions on Knowledge and Data Engineering, vol. 28, no. 12, pp. 3246 - 3260, 2016.

2.S. Sadik and L. Gruenwald, “Research issues in outlier detection for data streams,” ACM SIGKDD Explorations Newsletter, vol. 15, no. 1, pp. 33–40, 2014.

3.V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: Asurvey,” ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009.

4.S. Rajasegarar, C. Leckie, and M. Palaniswami, “Anomaly detection in wireless sensor networks,” IEEE Wireless Communications, vol. 15, no. 4, pp. 34–40, 2008.

5.D. Pokrajac, A. Lazarevic, and L. J. Latecki, “Incremental local outlier detection for data streams,” in Computational Intelligence and Data Mining, 2007, pp. 504–515.

6.M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, “Outlier Detection for Temporal Data: A Survey,” IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 1, pp. 1–20, 2013.

7.C. C. Aggarwal, “Outlier Analysis,” 2013.

8.K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne, “Online unsupervised outlier detection using finite mixtures with discounting learning algorithms,” in SIGKDD, 2000, pp. 320–324.

9.C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, “A framework for clustering evolving data streams,” in VLDB, 2003, pp. 81–92.

10.F. Cao, M. Ester, W. Qian, and A. Zhou, “Density-based clustering over an evolving data stream with noise,” in SIAM Conference on Data Mining, 2006, pp. 328–339.

11.C. C. Aggarwal, “A segment-based framework for modeling and mining data streams,” Knowledge and information systems, vol. 30, no. 1, pp. 1–29, 2012.

12.I. Assent, P. Kranen, C. Baldauf, and T. Seidl, “Anyout: Anytime outlier detection on streaming data,” in Database Systems for Advanced Applications, 2012, pp. 228–242.

13.E. M. Knox and R. T. Ng, “Algorithms for mining distance-based outliers in large datasets,” in International Conference on Very Large Data Bases, 1998, pp. 392–403.

14.F. Angiulli and F. Fassetti, "Detecting distance-based outliers in streams of data," in ACM Conference on Information and Knowledge Management, 2007, pp. 811–820.

15.D. Yang, E. A. Rundensteiner, and M. O. Ward, "Neighbor-based pattern detection for windows over streaming data," in Advances in Database Technology, 2009, pp. 529–540.

16.V. Niennattrakul, E. Keogh, and C. A. Ratanamahatana, "Data editing techniques to allow the application of distance-based outlier detection to streams," in IEEE International Conference on Data Mining. IEEE, 2010, pp. 947–952.

17.M. Elahi, K. Li,W. Nisar, X. Lv, and H.Wang, "Efficient clusteringbased outlier detection algorithm for dynamic data stream," in nternational Conference on Fuzzy Systems and Knowledge Discovery, vol. 5, 2008, pp. 298–304.

18.M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in ACM SIGMOD, vol. 29, no. 2, 2000, pp. 93–104.

19.S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "Loci: Fast outlier detection using the local correlation integral," in International Conference on Data Engineering, 2003, pp. 315–326.

20. A..Pingale, k.v.Metre " A Review of Computing Hotistic Measures on MRCube using MapReduce" in Joumal of lnnovation in Electronics and Comtrunication.

21. A. J. Gosavi, k .v. Metre," Discovering High Utility Itemsets using Hybrid Approach" in International Journal on Recent and Innovation Trends in Computing and Communication.

22. H. D. Gangurde, k.v.Metre "Clustering based Feature Selection from High Dimensional data" in International Journal on Recent and Innovation Trends in Computing and Communication.

23. M. B. Thombare , k. v. Metre "Aggregation Environment for Query Optimization in Network Monitoring
" in International Journal on Recent and Innovation Trends in Computing and Communication Volume: 2 Issue: 6

24. P. E. Patel , k. v . Metre "Continuous Query Processing Of Dynamic Data Items In Network Aggregation Environment" in International Journal of Innovative Research and Advanced Studies (IJIRAS) Volume 4 Issue 2, February 2017.

25.A. Jadhav, k. v. metre "Efficient Clustering Based Local Outlier Detection Over High Dimensional Streaming Data"
7th post graduate conference of computer engineering cpgcon2018.

26. A. Jadhav,, k .v . Metre "A Review on Various Outlier Detection Techniques" International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307.

27. M. U. Kharat, R. P. Dahake, k. v. Metre, Feature Dimension Reduction for Content-Based Image Identification, Book Chapter (Clustering Techniques for Content-Based Feature Extraction From Image) , IGI Global, (2018), pp. 100-121.

28. M.B. Thombare , K. V. Metre," Query Optimization and Executionof Dynamic Data Items in Network Aggregation Environment"
Proceedings of Third Post Graduate Conference on "Computer Engineering" cPGCON ,Elsevier.