http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (2): 128-137, 2009
ISSN 1812-5638

Distance Based Outlier for Data Streams Using Grid Structure

[1,2,3]Manzoor Elahi, [1,2]Lv Xinjie, [2,3]M. Wasif Nisar and [1,2]Hongan Wang
[1]Intelligence Engineering Laboratory, Institute of Software, Chinese Academy of Sciences,
F07, No. 5 Building, 4 No., South Fourth Street, Zhong Guan Cun, Beijing, 100190, China
[2]Graduate University of Chinese Academy of Sciences, Beijing, 100049, China
[3]COMSATS Institute of Information Technology, Islamabad, Pakistan

Abstract: This study deals with grid-based outlier detection method which can figure out most outstanding outliers from a high speed datastreams. It is capable to find outliers even with the evolution of datastream where there is a chance that object properties may change with the time. Grid structure used in this study can help to save number of extra calculations in case of nearest neighbor queries and can provide a solid platform for applying distance based nearest neighbor approach for finding outliers. Proposed grid based method efficiently partition incoming stream into chunks and store these chunks one by one into a fixed width grid structure for further processing. Each chunk of stream is processed with the combination of fixed width grid structure and distance based nearest neighbor approach. Through efficient pruning of safe regions, proposed method only needs to operate over the candidate regions for finding outliers. This method takes into account both, local and global view of outliers and assign score to each detected outlier and does not sacrifice the correctness of its results for fast processing time. Proposed method can operate faster, need limited memory resources, having low computation cost and found to be highly efficient for data stream environment. Several experiments on real and synthetic datasets show the effectiveness of proposed method.

Key words: Outlier detection, data streams, grid, mining

INTRODUCTION

Recently, several data mining methods (Datar *et al*., 2002; Manku and Motwani, 2002) for a data stream are actively introduced. Researches on a data stream are motivated by emerging applications involving massive data sets such as customer click streams, telephone records, multimedia data and sets of retail chain transactions can be modeled as data streams. Accordingly, a data stream is defined as a massive unbounded sequence of data elements continuously generated at a rapid rate. Due to this reason, it is impossible to maintain all elements of a data stream. Consequently, data stream processing should satisfy the following requirements (Garofalakis *et al*., 2002). First, each data element should be examined at most once to analyze a data stream. Second, memory usage for data stream analysis should be confined finitely although new data elements are continuously generated in a data stream. Third, newly generated data elements should be processed as fast as possible to produce the up-to-date analysis result of a data stream, so that it can be instantly utilized upon request. To satisfy these requirements, data stream processing sacrifices the correctness of its analysis result by allowing some errors.

Currently, outlier detection has been seen as an important task in data mining that enjoys a wide range of applications such as detections of credit card fraud, criminal activity and exceptional patterns in databases. A well-quoted definition of outliers is the Hawkin-Outlier which first appeared in (Hawkins, 1980). This definition states that an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. Hawkin-Outlier is defined in an intuitive manner. Recently, there have been numerous research work in outlier detection and some efficient notions such as distance-based outliers (Knorr and Ng, 1998, 1999; Ramaswamy *et al*., 2000) and density-based local outliers (Breunig *et al*., 2000; Jin *et al*., 2001) have been proposed. Each of these definitions follows the spirit of the Hawkin-Outlier. This study emphasize the scheme of distance based nearest neighbor outliers detection and the cell statistics, which is used to indicate the local outlierness of objects in databases.

Corresponding Author: Manzoor Elahi, Intelligence Engineering Laboratory, Institute of Software, Chinese Academy of Sciences, F07, No. 5 Building, 4 No., South Fourth Street, Zhong Guan Cun, Beijing, 100190, China

Traditional methods for outlier detection can produce good results on stored static dataset. These methods cannot be applied to streaming data efficiently as these methods are suitable for the environment where the entire dataset is already available and algorithm can operate in more than one pass. A general framework for mining data streams need small constant time per record along with the minimum memory requirement, using at most one scan of data. As the nature of data stream is unbounded the problem of mining outlier in data streams is often performed based on certain times intervals, usually called windows.

Recently, some data mining methods (Angiulli and Fassetti, 2007; Pokrajac *et al*., 2007) for a datastream have been actively introduced. But these methods try to implement the same traditional algorithms over the datastreams, which seems to be very inefficient for unbounded and high speed datastream environment. In the traditional database environment there is a lot of research towards outlier detection. Among those distance-based approach first proposed by Knorr and Ng (1998) and later this approach is further extended in different scenarios. All these techniques are highly dependent on the parameters provided by the users and the wrong choice can affect the results. Later LOF (Breunig *et al*., 2000) and its extension (Jin *et al*., 2001) although useful one, the computation of LOF values for every data objects require a large number of nearest neighbor searches.

This study proposes a grid-based outlier detection algorithm for finding outliers in data elements of a datastream. To find outliers over a datastream, the distribution statistics of data elements in the data space of a datastream are carefully maintained and properties of each cell are updated continuously. By keeping the distribution statistics and applying nearest neighbor approach over the candidate cells proposed method can find out better outliers within limited utilization of CPU and memory resources. Proposed method use pruning of safe regions and reduce the cost of expensive computations to find nearest neighbor for an element. Initially, the multi-dimensional data space of a datastream is partitioned into a set of equal-size fixed number of cells. With the continuous arrival of new data elements, the properties of respective grid cell are updated and new data is stored. With the help of cell properties proposed method can figure out candidate cells for further computations and mark those cells as safe cells which do not contain abnormal points. Data in safe cells will be pruned of and rest of the expensive distance computations will be performed on the limited number of cells available.

METHODS FOR OUTLIER DETECTION

There is an abundant literature on outlier detection algorithms in recent years which can be categorized into four groups:

- Statistical approaches
- Distance based methods
- Profiling methods
- Model-based approaches

In statistical techniques (Barnett and Lewis, 1994; Domingos and Hulten, 2001; Charikar *et al*., 2003), the data points are typically modeled using a stochastic distribution and points are labeled as outliers depending on their relationship with this model. Distance based approaches (Aggarwal and Yu, 2001; Breunig *et al*., 2000; Knorr and Ng, 1998) detect outliers by computing distances among points. Several recently proposed distance based outlier detection algorithms are based on:

- Computing the full dimensional distances among points using all the available features (Knorr and Ng, 1998) or only feature projections (Aggarwal and Yu, 2001)
- Computing the densities of local neighborhoods (Breunig *et al*., 2000; Papadimitriou *et al*., 2003)

In these methods the proposed notion of distance-based outliers, i.e., DB(pct, dmin)-outlier, which defines an object in a dataset as a DB(pct, dmin)-outlier if at least pct% of the objects in the datasets having the distance larger than dmin from this object. Later, the notion of distance-based outlier was extended and the distance to the kth nearest neighbors of a point p, denoted as Dk(p), is proposed to rank the point.

Distance-based methods do not rely on any assumed distribution to fit the data. Because they only examine the neighborhood for each object in the outlier detection, distance-based methods achieve better efficiency than the rest of the techniques within the domain of unsupervised learning. DB-Outliers has a limited ability in detecting outliers and cannot work well in some complex structure. In order to deal with the problems of simple distance based approaches, a density-based formulation scheme of outlier was proposed by Breunig *et al*. (2000). This formulation ranks the outlying degree of the points using Local Outlier Factor (LOF). Because LOF ranks points

only considering the neighborhood density of the points, thus it may miss the potential outliers whose densities are close to those of their neighbors. Jin *et al*. (2001) improved the efficiency of algorithm of Breunig *et al*. (2000) by proposing an efficient micro-cluster-based local outlier mining algorithm, but it still use LOF to mine outliers in dataset.

In addition, clustering-based techniques have also been used to detect outliers either as side products of the clustering algorithms (points that do not belong to clusters) or as clusters that are significantly smaller than others. However, since the main objective is clustering, they are not optimized for outlier detection. Furthermore, in most cases, the outlier definition or detection criteria are implicit and cannot easily be inferred from the clustering procedures. Some of the clustering approaches that can find outlier as a side product are CLARANS (Ng, and Han, 1994), DBSCAN (Ester *et al*., 1996), BIRCH (Zhang *et al*., 1997), WaveCluster (Sheikholeslami *et al*., 2000) and CLIQUE (Agrawal *et al*., 1998).

An Outlier detection technique proposed by (Aggarwal and Yu, 2001). The basic idea in their definition is, a point is an outlier, if in some lower dimensional projection it is present in a local region of abnormally low density. This method is also an efficient method for high dimensional data set. Some Clustering-Based outlier detection techniques are proposed by Eskin *et al*. (2002) and Arning *et al*. (1996). This technique study in two basic steps, fixed width clustering with w (radius) and after the first phase the next step is sorting of clusters produced in the first step. Points in the smaller clusters are declared as outliers. Clustering-based techniques are further extended by proposing the concept of cluster-based local outlier, in which a measure for identifying the outlierness of each data object is defined. Deviation-based techniques identify outliers by inspecting the characteristics of objects and consider an object that deviates from these features, declared as an outlier (Arning *et al*., 1996). Distance-based methods earlier discussed are designed to work in the batch framework, under the assumption that the whole data set is stored in secondary memory and multiple passes over the data can be accomplished. Hence, they are not suitable for data streams.

DISTANCE-BASED OUTLIER DEFINITION

Some of existing study in outlier detection lies in the field of statistics. Intuitively, outlier can be defined as given by Hawkins.

Definition 1: (Hawkins-Outlier): An outlier is an observation that deviates so much from other observations as to arouse suspicion that it is generated by a different mechanism.

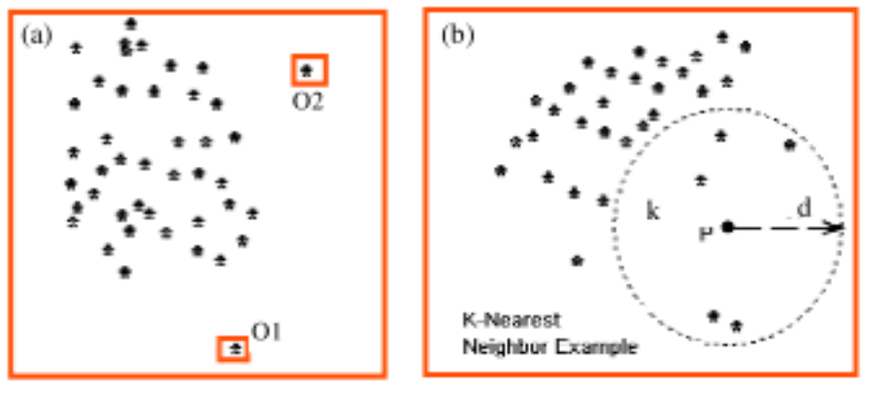

Fig. 1: Distance based outlier example (a) Dset-1 and (b) Dset-2

Definition 2: (DB(pct, dmin)-Outlier): An object p in a dataset D is a (DB (pct, dmin)−Outlier) if at least percentage pct of the objects in D lies greater than distance dmin from p, i.e., the cardinality of the set {q∈D|d(p, q) ⇐ dmin} is less than or equal to (100−pct)% of the size of D.

The definition of DB-Outlier studied here is the same as Knorr and Ng's work (Knorr and Ng, 1998). It is easy to conclude that the concept of DB-Outlier is well defined for any dimensional dataset. The parameter p stands for the minimum fraction of objects in a data space that must be outside an outlier's D-neighborhood. For ease to explain, proposed method employs another parameter M (M = N(1−p), N: data size) to represent the maximum portion of data points that must be in an outlier's D-neighborhood. It means that an outlier needs to have M or fewer objects within its D-neighborhood.

Figure 1a, b show two datasets, Dset-1 and Dset-2. Dset-1 shows two points O1 and O2 as most outstanding outliers on the data surface. These points are far away from the rest of the data points in the given dataset. Dset-2 shows an example of a distance based approach where the points P is shown as an inlier because P satisfy the criteria of containing K number of neighbors in this case 5 in a specified distance d where, K = 4.

This study use the definition 2 for finding DB-Outliers over the candidate cells.

PROPOSED GRID-BASED METHOD FOR DATASTREAM

It is assumed that, the number of outliers in any dataset is expected to be extremely small as compared to the normal data. So, it is highly inefficient to apply the traditional outlier detection algorithms over the entire data set, specially in case of datastream this method can become highly expensive as well as can often led us to wrong decision in finding most outstanding outliers.

Figure 2 shows some of the problems in existing methods. O1, O2, O3 are outliers within the cells. Although these points belong to a dense cell but within

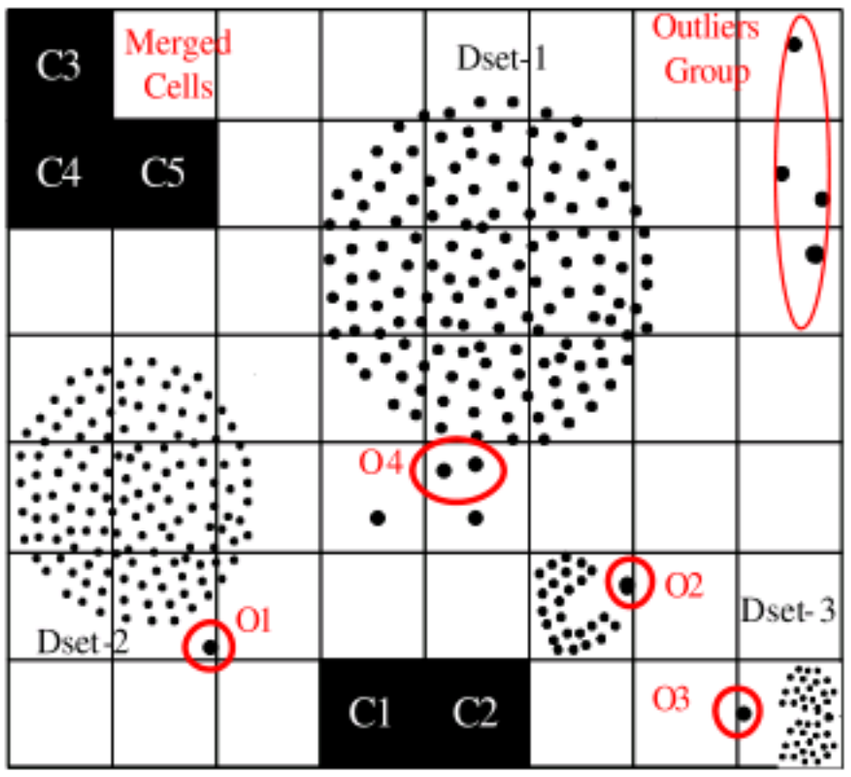

Fig. 2: Basic idea of grid based method

the cell, these points are far from the rest of the points. According to the distance based definitions of outliers these points should be declared as outliers. Most of the cluster based and grid based methods will declare point O4 as an outlier because it belongs to a very sparse cell which contains few points. In real sense these points has enough number of neighbors in the neighboring cells and should be treated as inliers. According to this observation, this study introduce an efficient grid-based outlier detection algorithm which help us to prunes away the portion of dataset which is safe and known to be non-outliers, In later stage with the help of nearest neighbor approach proposed study can find the points which are different from rest of the data points. Hence, the overall cost for computing is reduced. Instead of applying the distance based nearest neighbor over the entire available data, proposed method use a simple grid based structure to prune out the safe regions.

It is assumed that data stream objects are elements of a metric space on which is defined a distance function. At first, each dimension of the data space logically divided into fixed number of equi-width intervals, resulting in a grid like structure. As the grid is applied over the surface where the distance function can be used to find the relationship of points, incoming data can be efficiently accommodated in the exiting fixed grid. When all the exiting data is efficiently stored in the existing grid, it becomes easy to find the number of elements exists in each cell, which can also give us a measure of density of each cell. Some of the existing methods prune out the data available in dense cells and only consider the cell which contains lesser number of elements, for further processing for finding outliers. This approach can often lead us to a wrong decision, because it is possible that, some outlier may exist in the dense cells as well. Only the density measure of a cell can not guarantee that, a cell is free to contain any outlier candidates. Proposed method apply nearest neighbor approach over the grid cells to figure out the points different than the rest of the data in the same cell as well as different with respect to their neighboring cells. It is important to take into account the neighboring cells as well because in different experiments it is found that some border points in one cell seems more similar to the points exist in their neighbor cells. To handle this issue efficiently proposed method keep and update an average value μ_m^j for each cell as the new data point assigned to it. If a point is twice as far as the μ_m^j value of its respective cell as well as neighboring cells then that entire cell will be kept for further processing.

Proposed method only declare these points as a candidate outliers and compare them with next incoming stream chunk to make sure that these are real outliers.

Those grid cells which found to be safe will be pruned out while the candidate cells will still be kept there for further processing. Proposed approach do not suffer from exponential growth problem of the grid cells as the dimensionality increases as well as it also do not suffer from high computational cost, which is always a problem when distance based approaches are applied over high speed huge volume of datastreams.

Given a data stream D of d-dimensional data space, let's assume that datastream objects are elements of a metric space on which we can define a distance function.

$$DS = N_1 \times N_2 \times ... \times N_d$$

Consider a data stream as chunks of data $W = w_1, w_2,..,w_d$.

A data element generated at the jth chunk is denoted by $x^j = < x_1^j, x_2^j, x_2^j, ..., x_2^j, x_i^j \in N_i \; x_i^j \in N_i$.

The similarity between two data elements is defined as, the elements in the same cell or in some group of safe and dense neighboring cells.

Density of a cell is measured by the support of a cell; i.e., number of points belongs to that particular unit cell or group of cells.

Suppose x^j as a current chunk of stream, then sp_m^j and μ_m^j is defined as:

sp_m^j = No. of data elements in c_i^j
μ_m^j = Average of data elements in c_i^j. Calculated as:

$$\mu_m^j = \sum_{k=1}^{q} p_k$$

where, c_i^j denotes ith cell in jth chunk of stream.

When a new data element x_i^j arrives from the current chunk of the datastream x^j, its corresponding initial cell c_i^j

statistic is updated, i.e., the total number of data elements, average value.

The density of c_i^j is defined as the number of points contained in it, denoted as sp_m^j. A unit c_i^j is called non-empty if $sp_m^j > 0$, or dense if $sp_m^j \geq \delta$, where, δ is a density threshold. Proposed method consider the non empty Grid-Cells which are not dense as well as the dense cells containing some boundary elements and the dense grid cells having some local candidate outliers found after nearest neighbor quarries applied over each individual cell separately, while rest of the safe dense cells are pruned out from the grid.

During the second phase distance-based nearest neighbor strategy is applied over the remaining grid cells to efficiently figure out the outliers from each cell and later these outliers are assigned score on the basis of their deviation from the normal data within the cell, deviations from the neighboring cells, density of the cell from which the candidate outlier $CandOut_i^j$ belongs to and number of chunks L in which a $CandOut_i^j$ compared with the incoming data. Proposed method do not declare a point as an outlier just by comparing in one chunk of the datastream, it keep and compare the candidate outliers in user defined L number of stream chunks to assure the outlierness of these candidate outliers find in previous chunks.

Algorithm description: Basically proposed algorithm can be divided in 3 steps.

- Partition the space in equi-width grid cells and calculate the cell statistics
- Figure out candidate cells; merge cells which are dense and safe from containing candidate outliers. Later pruned out these cells
- Apply distance based outlier detection algorithm over candidate cells
- After appropriate number of stream chunks (L), declare candidate outliers as real outliers or inliers and assign outlierness score to the final outliers

Algorithm 1: Mining Stream outlier using GRID

Require: n: No. of grid cells
Require: L: No. of Iterations for candidate point for outlierness degree
Require: W = $w_1, w_2,...,w_d$: Chunks of data stream
Require: k: No. of nearest neighbors
Require: δ: Density threshold
Step 1: Create n number of grid cells
Step 2: For each w_1
(i) Assign x_i^j to its appropriate cell.
(ii) Update the properties of cell c_i^j
(a) sp_m^j = No. of data elements in c_i^j
(b) μ_m^j = Average of x_i^j in c_i^j
Step 3: Merge dense neighboring cells using density threshold δ
Step 4: If $(x_i^j > \mu_m^j)$
Mark cell c_i^j as candidate cell
Step 5: Prune of the safe regions
Step 6: For each candidate c_i^j
(i) Apply Db-Outlier over the cell.
(ii) Keep the $CandOut_i^j$ while discard rest of the data
Step 7: Move $CandOut_i^j$ to the next chunk of stream
Step 8: Assign outlierness degree to each detected outliers using, density measure and deviation value in its respective cell and neighboring cells

EXPERIMENTAL STUDY

A comprehensive performance study has been conducted to evaluate proposed algorithm. The algorithm was run on both reallife datasets obtained from the UCI machine learning repository and synthetic datasets. The experimental results show that in each case proposed algorithm is very efficient for high speed unbounded datastreams not only in finding most outstanding outliers but also efficient regarding CPU utilization and memory usage.

In order to show the performance of the proposed Gridbased approach, first test of the proposed method is to show the accuracy of finding most outstanding outliers in large datasets. Synthetic dataset is generated using the synthetic data generating system (Pokrajac *et al*., 2007). For better visual inspection experiments of the proposed method are based on two-dimensional synthetic data. The sizes of these datasets range from (500 to 10,000) data points.

Based on the fact that outliers accounts for only a very small portion in a dataset, the proposed algorithm efficiently figure out the areas which expected to contains outliers while discard those areas which are safe from containing outliers. To show the performance of the proposed grid-based approach for finding outliers, first test results of this study are shown in Fig. 3-6. Proposed algorithm is tested for different sizes of the data. Figure 3-6 are based on 500, 2000, 2500 and 3000 data records, respectively.

These results can show the accuracy of the proposed approach for finding outliers. In the proposed method stream of data is always divided into chunks and each chunk examined by further grid processing overall final outliers produced by the proposed method are shown in Figure 3-6 based on synthetic dataset.

Figure 7 and 8 show the test results of the proposed method over the reallife datasets. Test results of Fig. 7

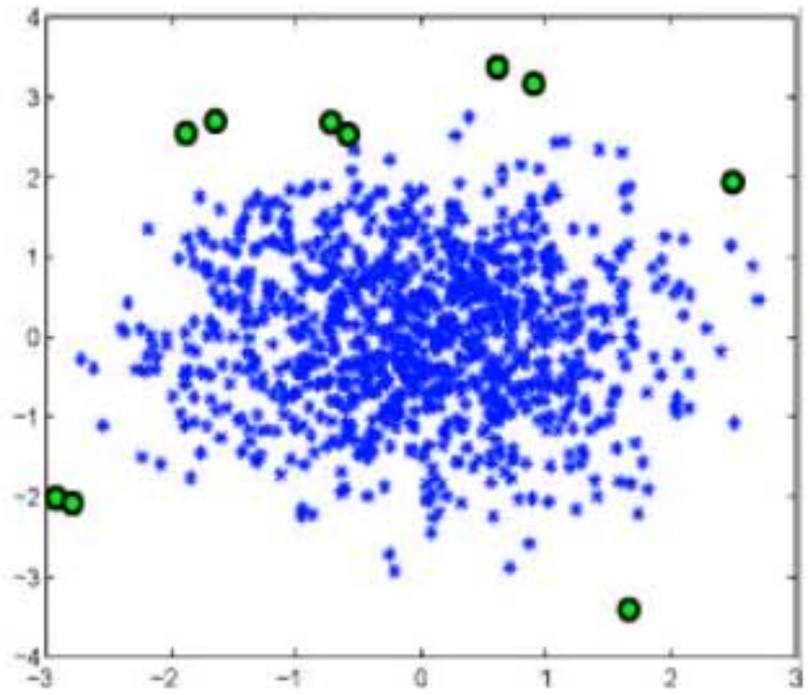

Fig. 3: Synthetic dataset (N = 500)

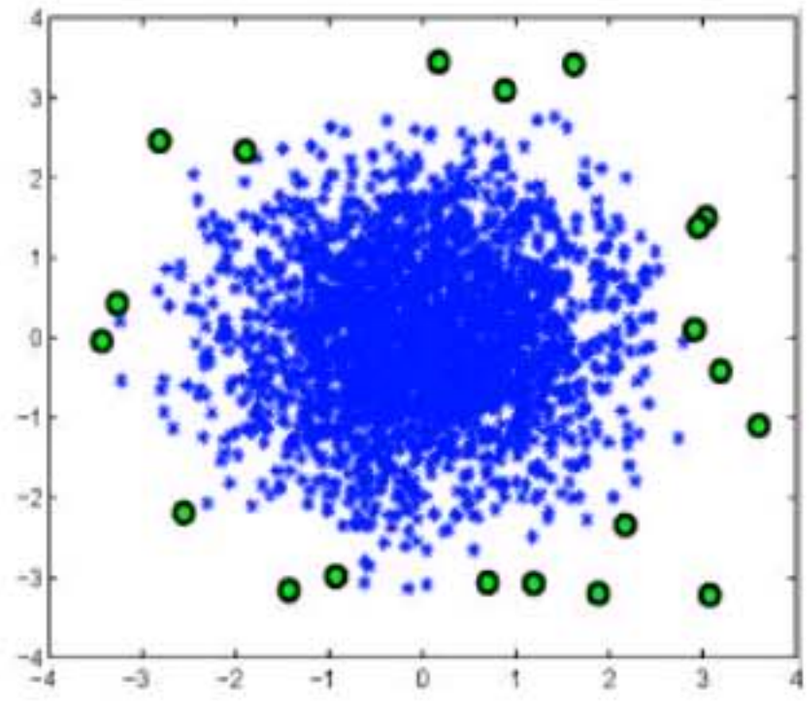

Fig. 4: Synthetic dataset (N = 2000)

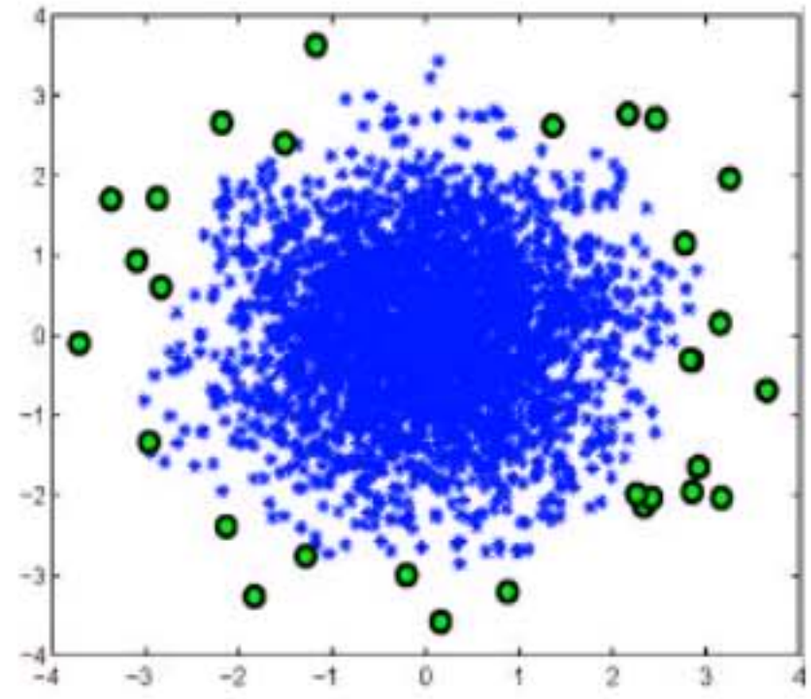

Fig. 5: Synthetic dataset (N = 2500)

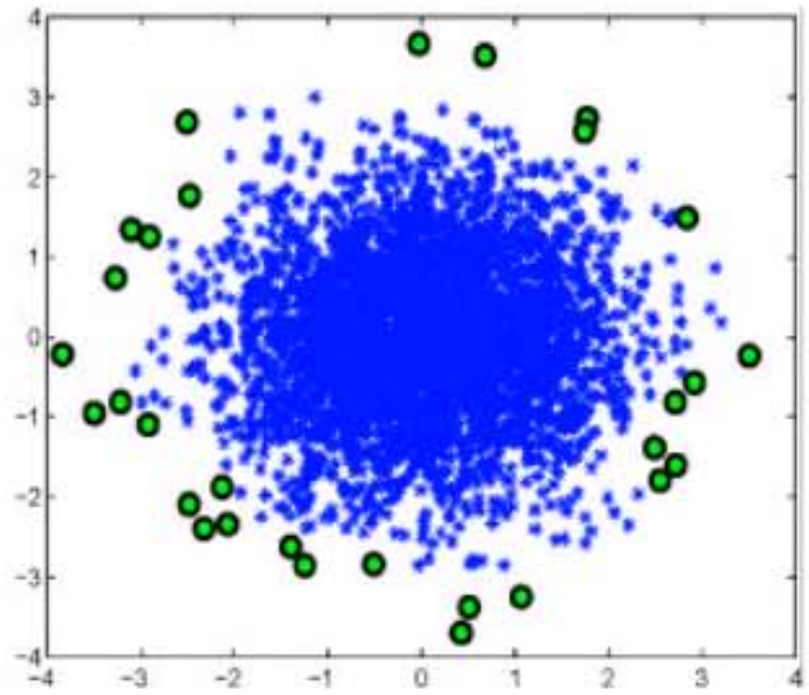

Fig. 6: Synthetic dataset (N = 3000)

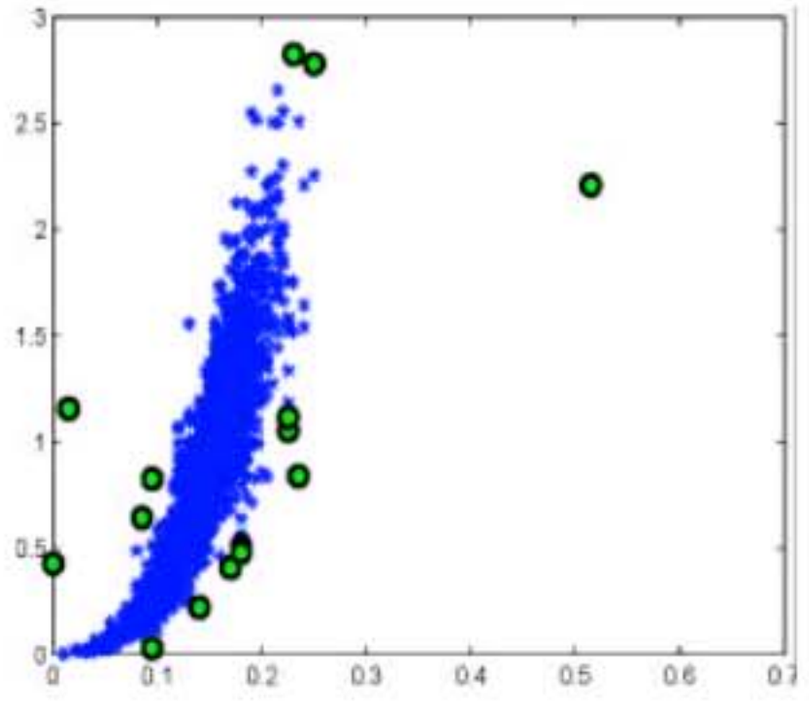

Fig. 7: RealLife abalone dataset

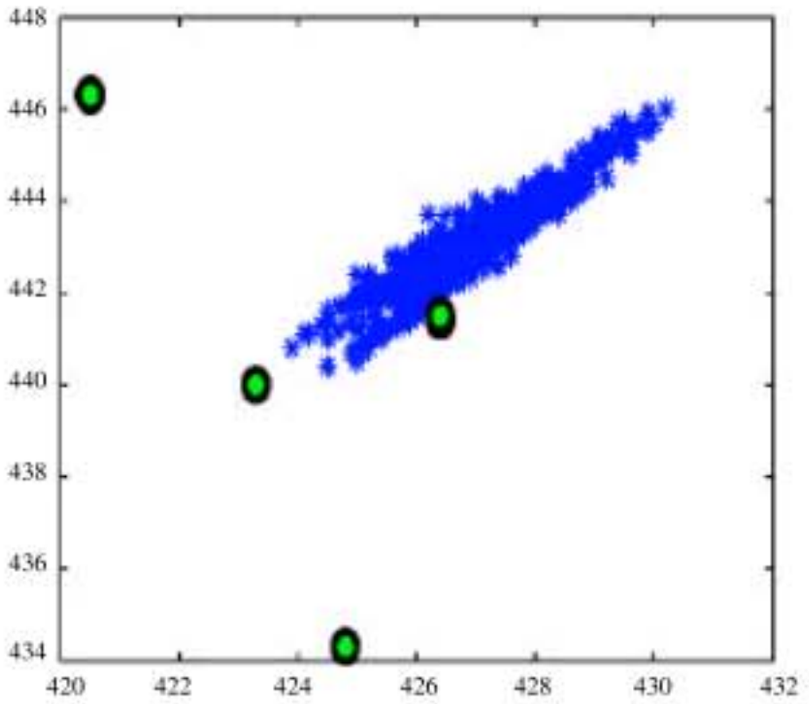

Fig. 8: RealLife temperature dataset

consists of Abalone dataset obtained from UCI machine learning repository (Asuncion and Newman, 2007). Figure 8 shows the test results of a dataset obtained from temperature device operating in an industry. All these test results can prove the accuracy of the proposed method in finding suitable outliers.

TIME AND SPACE COMPLEXITY ANALYSIS

Time complexity of the proposed algorithm is much better than the existing approaches. Almost all the existing methods proposed for datastream take into account the entire available stream for finding outliers. Nearest neighbor quarries in case of directly applying distance based or density based methods over entire available dataset can tend be inefficient in CPU utilization as well as regarding pace with which the algorithm need to work over the high speed data. Proposed algorithm efficiently prune of the safe cells and save huge number of extra calculations.

Figure 9 shows the difference between the proposed Grid-based approach and DB-outlier as a whole. Through number of experiments it is observed that the efficiency of the proposed grid-based method increased over existing direct approaches, with the increase in size of dataset.

The time complexity varies with the different values of parameter K which denote the number of nearest neighbor required for a point to be declared as inlier or outlier. First comparison is done for different values of K within each approach to see the effect of K over the CPU utilization. Figure 10 shows the effect of parameter K over the existing DB-outlier as whole, while Fig. 11 shows the test results of proposed method. It can be seen that, there is little effect over CPU utilization with the increase in value of K in case of proposed method, while more in case of directly applying DB-outlier over the available dataset.

Figure 12-14 show the detailed comparison between proposed method and DB outlier as a whole with different size of the dataset. Each of these results can prove the efficiency of the proposed method with the increase in size of dataset. Because in the proposed method the data is divided into chunks and it efficiently prune out the safe regions which help to reduce the time complexity and to make the proposed method feasible to keep pace with high speed datastreams.

Proposed algorithm is also very efficient regarding space complexity. Proposed method can work in limited memory resources as the incoming stream is divided into fixed number of chunks. There is no need to accommodate the entire available stream data into the memory. Some algorithms for mining streams need to have low memory

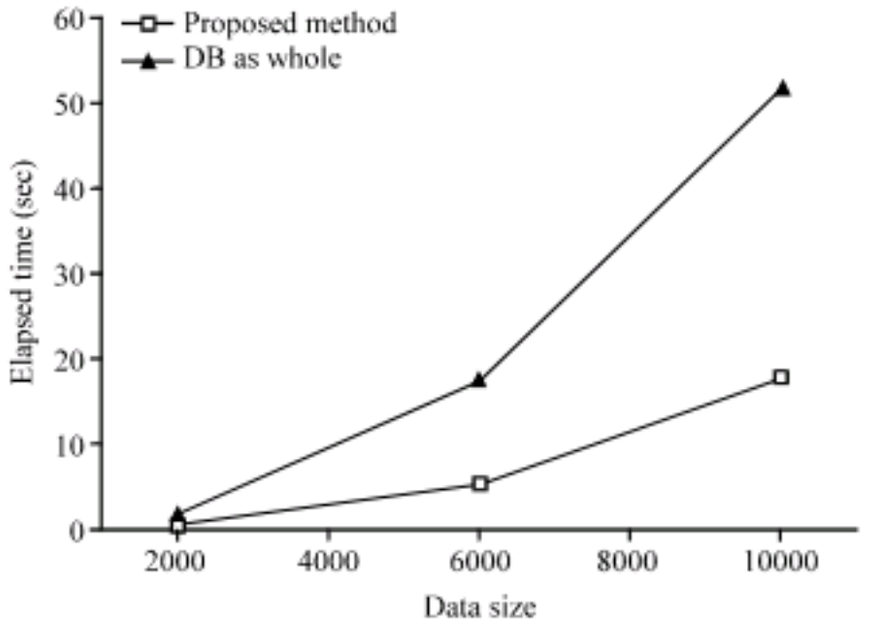

Fig. 9: Time complexity analysis proposed vs DB as a whole

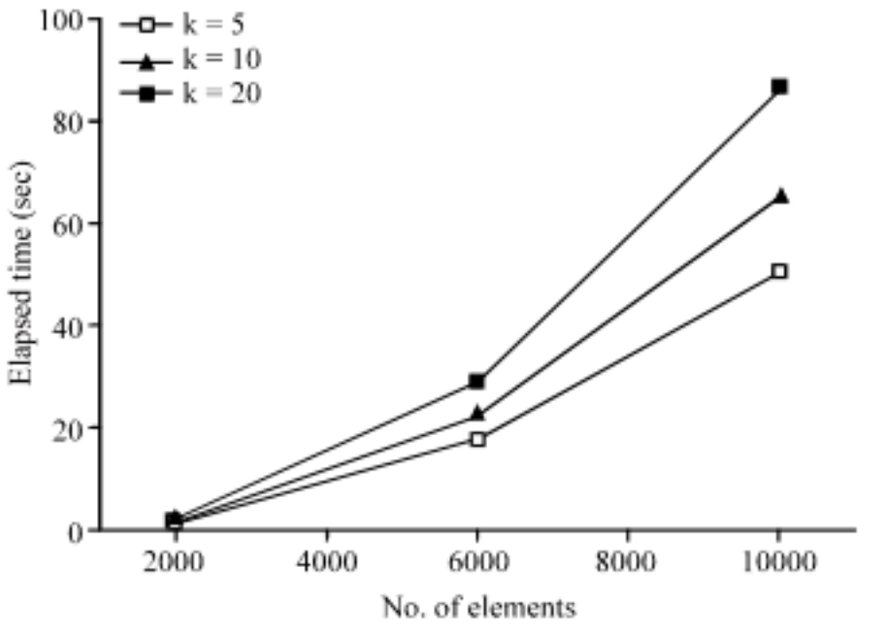

Fig. 10: Effect of K over DB as a whole

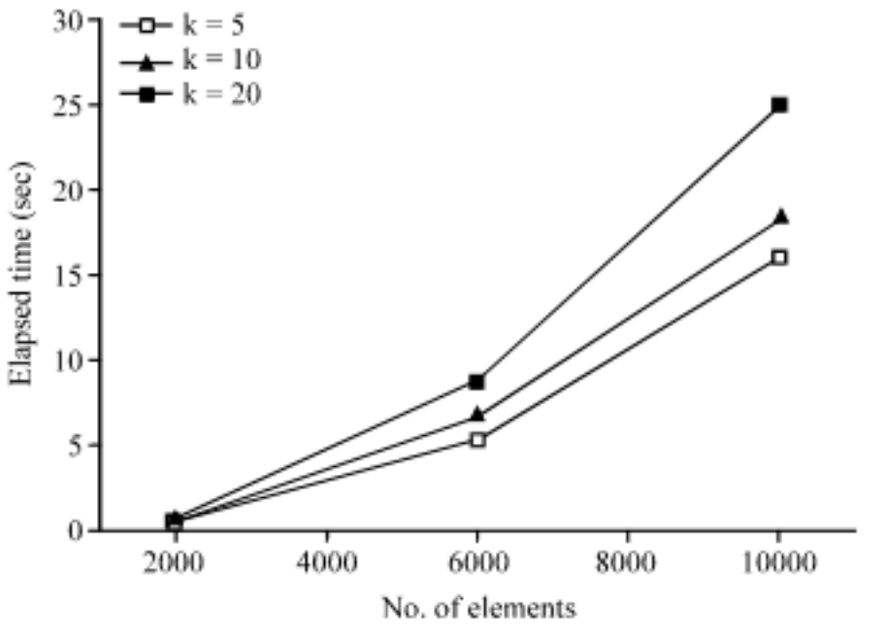

Fig. 11: Effect of K over proposed method

requirements, as stream mining often carried out in small and hand held devices that do not have much memory. Figure 15 shows the comparison of memory usage between the proposed method and applying distance based nearest neighbor approach as a whole over the different size of dataset like, 2, 4, 6 and 10 k. These results

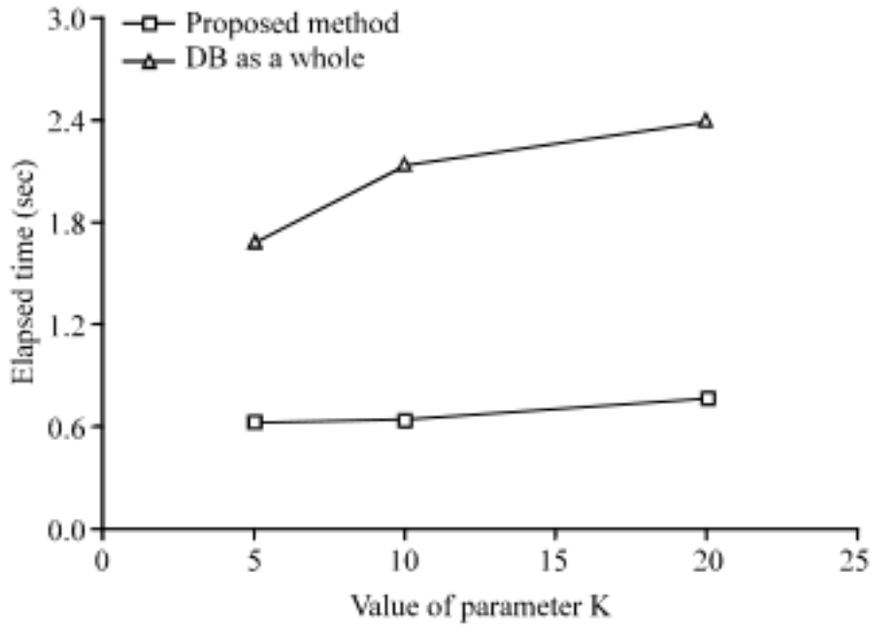

Fig. 12: Effect of K with N = 2000

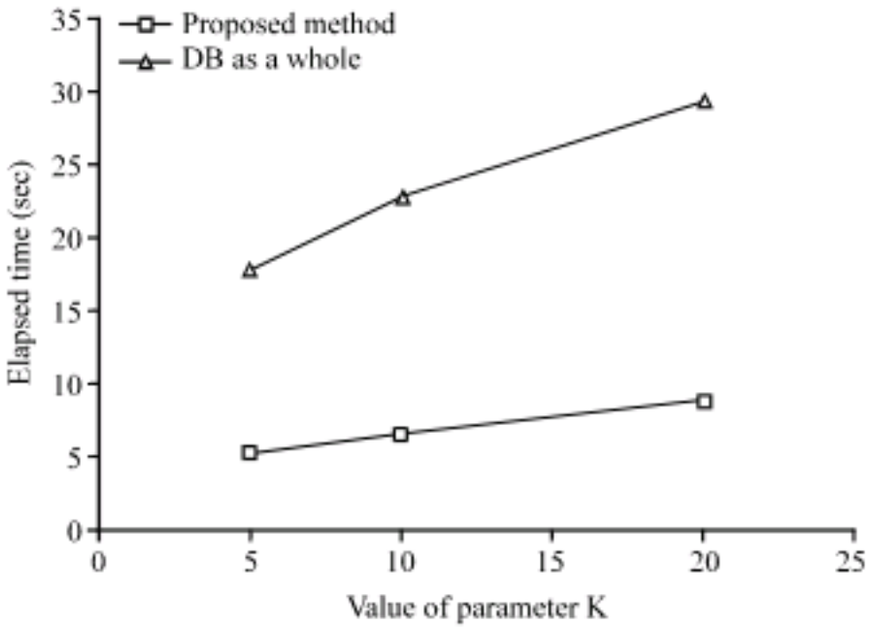

Fig. 13: Effect of K with N = 6000

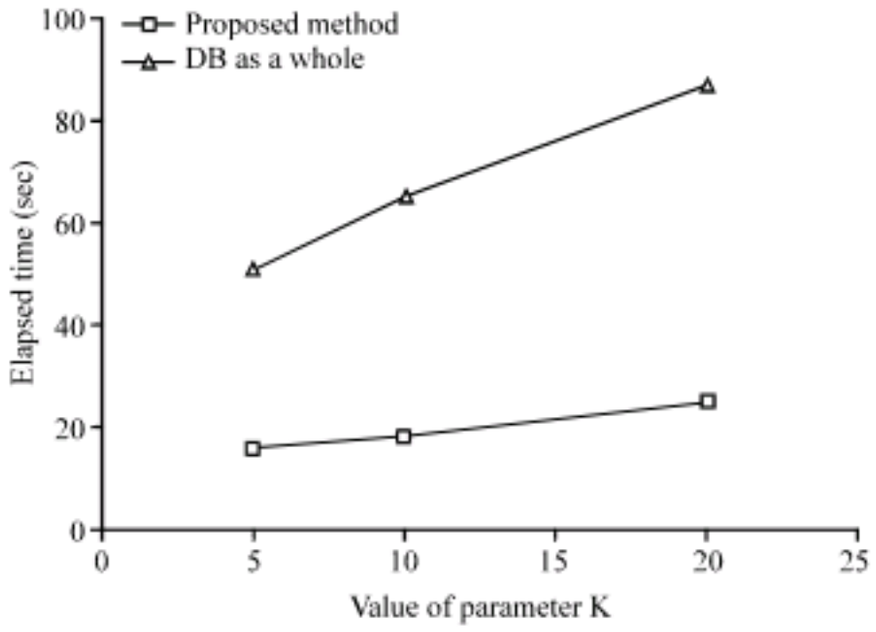

Fig. 14: Effect of K with N = 10,000

can prove that the proposed method has much better space complexity compared to the existing methods and can be efficient for high speed datastreams.

The proposed method work like an incremental fashion. Points which are declared as candidate outliers in

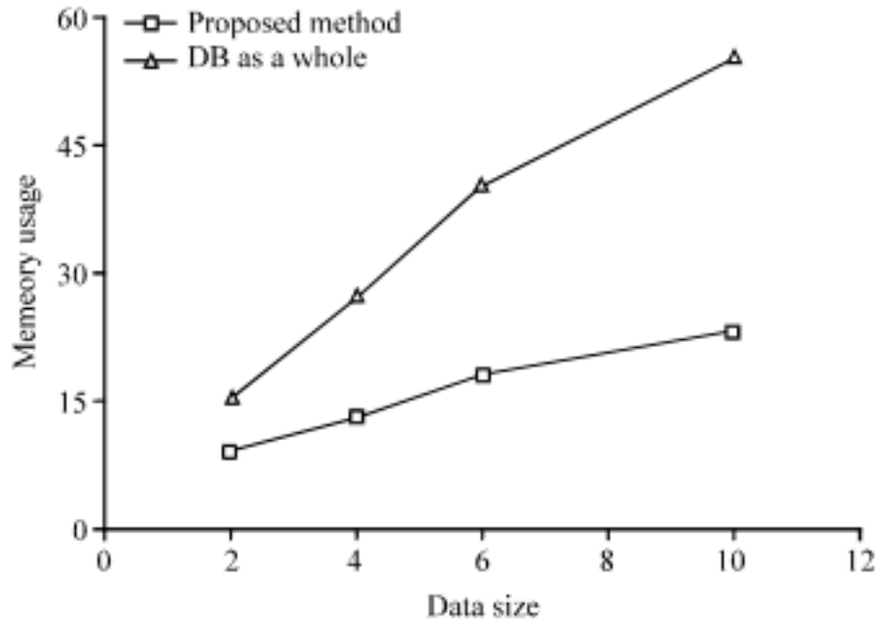

Fig. 15: Comparison of memory usage for 2, 4, 6 and 10 k data size over proposed vs. DB-Outlier as a whole

one chunk of the datastream will be compared with next predefined number of chunks. Most of the methods declare a point as an outlier as it arrives. Although this technique is efficient for some scenarios but often can led us to wrong decision because of the dynamic nature of the datastreams. Object properties may change with the evolution of datastream, points detected as outlier in current stream chunk may become an inlier as the new data is processed. In order to handle this issue proposed method work with a user defined parameter L to check the validity of the detected outliers.

CONCLUSION AND FUTURE RESEARCH

Outlier detection is currently very active area of research in datastream mining community. However, earlier research for the problem of outlier detection is suitable for disk resident datasets where the entire dataset is available in advance and algorithms can operate in more than single passes.

Most of the existing work for outlier detection over the datastream only focus on detection rate of outliers while ignoring the most important issue of data stream mining like, low memory requirements and high speed algorithms to keep pace with high speed unbounded datastreams. In this study, an outlier detection algorithm which uses grid structure with the combination distance-based outlier detection technique is proposed. Experiments have been carried out on both real data and synthetic data. The experiments show that the proposed algorithm is more suitable for datastream environment and can figure out suitable outliers and assign degree of outlierness in limited space and time complexity. There are several opportunities for future research. Fixed grid can

be converted into a dynamic gird with efficient data dimensionality reduction technique like manifold learning for very high dimensional datasets. Pure density based methods like LOF can be optimized to use with grid structure. Nearest neighbor findings can be improved over the each grid cells by effective distance function for high-dimensional data.

ACKNOWLEDGMENTS

This research is jointly supported by COMSATS Institute of Information Technology under Faculty Development Program and National High-Tech Research and Development Plan of China under Grant No. 2006AA042182 by Institute of Software Chinese Academy of Sciences.

REFERENCES

Aggarwal, C.C. and P. Yu, 2001. Outlier detection for high dimensional data. Proceedings of the ACM SIGMOD International Conference on Management of Data, May 21-24, ACM New York, USA., pp: 37-46.

Agrawal, R., J. Gehrke, D. Gunopulos and P. Raghavan, 1998. Automatic subspace clustering of high dimensional data for data mining applications. ACM SIGMOD Rec., 27: 94-105.

Angiulli, F. and F. Fassetti, 2007. Detecting distance-based outliers in streams of data. Proceedings of the 60th ACM conference on Conference on Information and Knowledge Management, Nov. 6-10, ACM New York, USA., pp: 811-820.

Arning, A., R. Agrawal and P. Raghavan, 1996. A linear method for deviation detection in large databases. Proceedings of the International Conference on Knowledge Discovery and Data Mining, 1996, Portland OR., USA., pp: 164-169.

Asuncion, A. and D.J. Newman, 2007. UCI machine learning repository. University of California, Department of Information and Computer Science, Irvine, CA. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Barnett, V. and T. Lewis, 1994. Outliers in Statistical Data. 3rd Edn., John Wiley and Sons, New York, ISBN: 0-471-93094-6, pp: 584.

Breunig, M.M., H.P. Kriegel, R.T. Ng and J. Sander, 2000. LOF: Identifying density based local outliers. ACM SIGMOD Rec., 29: 93-104.

Charikar, M., L. OCallaghan and R. Panigrahy, 2003. Better streaming algorithms for clustering problems. Proceedings of the 35th Annual ACM Symposium on Theory of Computing, Jun. 9-11, ACM New York, USA., pp: 30-39.

Datar, M., A. Gionis, P. Indyk and R. Motwani, 2002. Maintaining stream statistics over sliding windows. SIAM J. Comput., 31: 1794-1813.

Domingos, P. and G. Hulten, 2001. A general method for scaling up machine learning algorithms and its application to clustering. Proceedings of the 18th International Conference on Machine Learning (ICML), Jun. 28-Jul. 1, San Francisco, CA., USA., pp: 106-113.

Eskin, E., A. Arnold, M. Prerau, L. Portnoy and S. Stolfo, 2002. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Data Mining for Security Applications. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.5445.

Ester, M., H.P. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Han, J. and U.M. Fayyad, (Eds.). Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996, AAAI Press, pp: 226-231.

Garofalakis, M., J. Gehrke and R. Rastogi, 2002. Querying and mining data streams: You only get one look. Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Jun. 3-6, ACM New York, USA., pp: 635-635.

Hawkins, D.M., 1980. Identification of Outliers. 1st Edn., Chapman and Hall, London, New York, ISBN-13: 9780412219009.

Jin, W., A.K.H. Tung and J. Han, 2001. Mining top-n local outliers in large databases. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 26-29, ACM New York, USA., pp: 293-298.

Knorr, E.M. and R.T. Ng, 1998. Algorithms for mining distance-based outliers in large dataset. Proceedings of the 24th International Conference on Very Large Data Bases, Aug. 24-27, San Francisco, CA, USA., pp: 392-403.

Knorr, E.M. and R.T. Ng, 1999. Finding intentional knowledge of distance-based outliers. Proceedings of the 25th International Conference on Very Large Data Bases, Sept. 7-10, San Francisco, CA., USA., pp: 211-222.

Manku, G.S. and R. Motwani, 2002. Approximate frequency counts over data streams. Proceedings of the 28th international Conference on Very Large Databases, Aug. 20-23, ACM New York, USA., pp: 346-357.

Ng, R.T. and J. Han, 1994. Efficient and effective clustering methods for spatial data mining. Technical Report: TR-94-13.http://portal.acm.org/citation.cfm?id=901953.

Papadimitriou, S., H. Kitagawa, P.B. Gibbons and C. Faloutsos, 2003. LOCI: Fast outlier detection using the local correlation integral. Proceedings of the 19th International Conference on Data Engineering (ICDE'03), Mar. 5-8, Pittsburgh, PA., USA., pp: 315-326.

Pokrajac, D., A. Lazarevic and L.J. Latecki, 2007. Incremental local outlier detection for data streams. Proceedings of the Symposium on Computational Intelligence and Data Mining, Mar. 1-Apr. 5, Honolulu, HI., pp: 504-515.

Ramaswamy, S., R. Rastogi and S. Kyuseok, 2000. Efficient algorithms for mining outliers from large data sets. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 15-18, ACM New York, USA., pp: 427-438.

Sheikholeslami, G., S. Chatterjee and A. Zhang, 2000. Wavecluster: A wavelet-based clustering approach for spatial data in very large databases. The VLDB J., 8: 289-304.

Zhang, T., R. Ramakrishnan and M. Livny, 1997. Birch: A new data clustering algorithm and its applications. Data Mining Knowledge Discovery, 1: 141-182.