Unsupervised Outlier Detection in Streaming Data Using Weighted Clustering

Yogita, and Durga Toshniwal

Abstract—Outlier detection in streaming data is very challenging because streaming data cannot be scanned multiple times and also new concepts may keep evolving. Irrelevant attributes can be termed as noisy attributes and such attributes further magnify the challenge of working with data streams. In this paper, we propose an unsupervised outlier detection scheme for streaming data. This scheme is based on clustering as clustering is an unsupervised data mining task and it does not require labeled data, both density based and partitioning clustering are combined for outlier detection. In this scheme partitioning clustering is also used to assign weights to attributes depending upon their respective relevance and weights are adaptive. Weighted attributes are helpful to reduce or remove the effect of noisy attributes. Keeping in view the challenges of streaming data, the proposed scheme is incremental and adaptive to concept evolution. Experimental results on synthetic and real world data sets show that our proposed approach outperforms other existing approach (CORM) in terms of outlier detection rate, false alarm rate, and increasing percentages of outliers.

Keywords—Concept Evolution, Irrelevant Attributes, Streaming Data, Unsupervised Outlier Detection.

I. Introduction

NOWADAYS many applications are generating streaming data for an example real-time surveillance, medical systems, internet traffic, online transactions and remote sensors. Data streams and streaming data are synonymous. Data streams are temporally ordered, fast changing, massive, and potentially infinite sequence of data objects [1]. Unlike traditional data sets, it is impossible to store an entire data stream and to scan through it multiple times due to its tremendous volume. New concepts may keep evolving in data streams over time. Evolving concepts require data stream processing algorithms to continuously update their models to adapt to the changes.

Outlier detection is a data mining task. It is also known as outlier mining. An outlier is an object that does not comply with the behaviour of normal data objects. In many applications outliers are more interesting than normal cases for example network intrusion detection, fault diagnosis in machines (motors, space shuttles, etc.), credit card fraud detection, marketing, detecting outlying cases in wireless sensor network data.

It is very difficult to collect labelled data for data mining and also new concept may come to existent and others may get outdated in streaming data. In such a scenario unsupervised data mining approaches are more feasible as compare to supervised approaches. Unsupervised outlier detection approaches do not require class labels of objects and can detect unforeseen outlying cases. Clustering based outlier mining methods are unsupervised in nature; do not require knowledge of data in advance. Density based clustering methods can produce outlying objects along with normal cluster. Partitioning based clustering methods can be used for distance based outlier detection.

Most of existing clustering based outlier detection methods give equal importance to relevant and irrelevant (noisy) attributes (In the present paper, irrelevant attributes are taken synonymous with noisy attributes). This leads to their poor performance on real world data having noisy attributes. Because performance of clustering based outlier detection method depends on the quality of clusters discovered. Presence of noisy attributes conceals real clustering structure of data and hence leads to lower outlier detection rate and higher false alarm rate [2].

In this paper we have proposed a clustering based unsupervised outlier detection scheme for streaming data. In the proposed approach both density based and partitioning clustering are combined to take advantage of both density based and distance based outlier detection. This scheme assigns weights to attributes depending upon their respective relevance in mining tasks using weighted k-mean clustering [2]. As oppose to most of existing attribute weighting methods that use static attribute weights, keeping in view the streaming data environment weights are updated periodically to adapt them according to evolving concepts.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 describes proposed scheme. Section 4 presents experimental results. Section 5 concludes the paper.

II. Related Work

Outlier detection has been very interesting topic for research community [3]–[11]. Ramaswamy et al proposed a distance based outlier detection method in [3]. According to which, given parameters k and n, an object is an outlier if no more than n-1 other objects in the dataset have higher value for D_k than object o, where D_k(o) denotes the distance of k^{th} nearest neighbor of object o. This idea is further extended in [10], where each data point is ranked by the sum of distances from its k^{th} nearest neighbors. Breunig et al introduced the notion of the local outlier factor LOF in [4], which captures the relative degree of outlierness of an object. Above described methods

Yogita and D. Toshniwal are with the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, Uttrakhand, 247667 India e-mail: (thakranyogita@gmail.com).

are either distance based or nearest neighbors based that are not suitable for outlier detection in data streams due to their high time complexity. He et al in [5] presented new definition of outlier which they named as cluster-based local outlier, which provides importance to the local data behaviour. Duan et al in [11] proposed a cluster based outlier detection algorithm which can detect both single point outliers and cluster-based outliers.

But all these method that we have described above and many more are proposed for stored static data sets and are not applicable in data streams environment. In order to resolve this problem, the exact-STORM and approx-STORM algorithms [7] are presented for detecting distance-based outliers in streams of data. Outlier detection techniques for data streams that are proposed in [7], [8] works upon the notion of sliding window. These techniques are very much dependent upon the window size selection. A Distance Based Outlier Detection for Data Streams (DBOD-DS) is proposed be Sadik et al in [9]. DBOD-DS detects outliers based on two user-defined parameters that are neighbor radius and minimum neighbour density. But DBOD-DS is unable to handle concept evolution in streaming data. Cluster based OutlieR Minera (CORM) is presented by Elahi et al in [6]. It is a clustering-based approach for outlier detection based on k-mean. It divides data stream in chunks of data for processing. Its performance is poor on grouped outlier as it treats those as normal data clusters. Yogita et al has proposed a framework for outlier detection in evolving data streams by weighting attributes in clustering [2]. It is a clustering based framework that assigns weights to all attributes depending on their respective relevance in clustering. Also the weights are updated periodically to adapt them according to evolving concepts.

III. Proposed Scheme Unsupervised Outlier Detection in streaming Data using Weighted Clustering

In this section proposed scheme is presented. Pictorial representation of proposed scheme is given in Fig. 1.

A. Preliminary Concepts

- Data Stream - A DataStream DS = x$_1$, x$_2$,.........,x$_n$ is a unbounded sequence of data objects. Object x$_i$ = (x$_{1i}$,x$_{2i}$,..........., x$_{mi}$) is characterize by a set of m attributes. We have processed data stream in form of data chunks. Every data chunk contains specified number of n points.
- Data Chunk - Stream of data is an unbounded sequence of data. As it is not possible to store complete data stream, for processing we divide it into data chunks of same size. Chunk size is specified by the user which depends upon the nature of data. In our scheme a object will be examined over multiple consecutive data chunks before declaring it as outliers because a cluster may be split over two data chunks or new cluster may be emerging and a outlying object of current data chunk may become inlier when similar data objects come in next chunks.
- Weight - Weight of an attribute gives its degree of importance or relevance in data stream mining. Large weights are assigned to relevant attribute and smaller to noisy attributes. Sum of all weights is always one. When we take weights in distance calculation we get good measure of distance. So all distance calculations in updation, clustering and outlier detection part of proposed scheme use corresponding attribute weights.
- Outlier Detection - Given a data stream DS, chunk size n, and weight vector w = (w$_1$, w$_2$, ...w$_m$) detection of outlier is to find objects which deviates from normal clusters and small in numbers until L number of chunks. These objects can be in groups or individual.
- Variance Matrix - Variance matrix is of $1 \times m$ dimensions where m is number of attributes. Each entry of this matrix stores sum of distance of all object from their corresponding cluster in corresponding attribute.
- Maximum Score (L) It specify till how many data chunks outlierness of an objects is tested before declaring it as real outlier [6].
- Candidate Outlier - An object is candidate outlier if it deviates more than the given threshold from normal clusters based upon deviation criteria. We have used same criteria as defined in [12] for scattered outliers because it is useful even when clusters are of different density and size.
- Real Outlier A candidate outlier becomes real outlier when it fulfils deviation criteria up to L data chunks.

B. Data Chunks & Data Pre-processing

Stream of data is an unbounded sequence of data. As it is not possible to store complete data stream, for processing we divide it into data chunks of same size. Chunk size is specified by the user which depends upon the nature of data. When current data chunk is processed during that time incoming data is stored in buffer and later taken out as data chunk. After storing required statistics, processed data chunk is deleted at the end of processing iteration to empty space for next incoming data chunk.

Real world data sets are highly susceptible to missing and inconsistent data. Such datasets are of low quality and leads to low quality mining results because quality of mining results depends upon the quality of data. This module applies data pre-processing techniques to improve data quality. Pre-processing techniques includes handling missing values, data scaling, aggregation, normalization, discretization etc. Which technique is applied depends upon the type data stream.

C. Density Based Clustering

In this section DBSCAN algorithm is used for clustering current data chunk. It does not require number of clusters and can find the arbitrary shape clusters. DBSCAN output is a set of clusters and outlying objects. Outlying objects are considered as candidate outliers (that have possibility of being real outlier) and input to outlier detection for further verification of their outlying nature. Small size clusters (For deciding small or large cluster criteria of [5] is taken) of

DBSCAN may be group of outliers or these may be portion of a cluster that yet to be come in next data chunk and has been split over to chunks. So these objects are also treated as candidate outliers and feed to outlier detection module. For clustering the current data chunk weights of previous phase are taken and that are then updated in weighted k-mean clustering module. DBSCAN parameters MinPts & epsilon are updated using equation (1) and (2)

$$\left.\begin{aligned} &\text{Epsilon} = \frac{\sum_{i=1}^{k} \text{Avg Intra } (Ci)}{k} \\ &\text{where}\quad \text{Avg Intra } (Ci) = \frac{\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} \text{Dist}\,(Oi,Oj)}{2\times n} \end{aligned}\right\} \quad (1)$$

$$\left.\begin{aligned} &\text{Min Pts} = \text{Avg No. of Objects with a distance of Epsilon} \\ &\qquad \text{from a object in cluster of smallest density} \\ &\text{where Density } (Ci) = \frac{\text{No. of Object } (Ci)}{\text{Radius } (Ci)} \end{aligned}\right\} \quad (2)$$

In above equations k is the number of clusters, n is the total number of objects comprises all clusters. C$_i$ represent ith cluster and o$_i$ represent ith object. Dist( ) is the distance between two objects. In our experiments we have taken Euclidian distance. Based on these equation parameters of DBSCAN are updated in MinPts & epsilon updation section.

D. Updation Module

Objects of DBSCAN clusters are assigned to their nearest cluster centres of previous phase clustering. For initial phase DBSCAN clusters are considered as it is and their mean are taken for centres. But in other phases after assigning objects to centres these newly assigned objects are used to calculate current centres and variance matrix. Updated centres are calculated using additive property, by summing up current and previous phase centres. Weights are calculated using equation (3) and (4)

$$\left.\begin{aligned} &\text{w}_j = \begin{cases} 0 & if\ D_j=0 \\ \frac{1}{\sum_{t=1}^{h}\left[\frac{D_j}{D_t}\right]^{\frac{1}{\beta-1}}} & if\ D_j\neq 0 \end{cases} \\ &where\text{Dj=}\sum_{l=1}^{k}\sum_{i=1}^{n} u_{i,l}\, d(x_{i,j}, z_{l,j}) \end{aligned}\right\} \quad (3)$$

$$\sum_{j=1}^{m} w_j = 1 \text{ and } 0 \leq w_j \leq 1 \quad (4)$$

In above equations w$_j$ represents weight of jth attribute and h is the number of variable where D$_j$ $\neq 0.\beta$ is the parameter supplied by user, k gives the number of clusters, n is the number of objects, u$_{i,l}$ is one if ith object is member of lth cluster otherwise it is zero, d is the distance measure like Euclidian or could be any other measure. Z$_l$ is center of lth cluster [13].

In our proposed approach to calculate updated weights D$_j$ is equal to sum of previous phase and current phase variance matrix entries corresponding to jth attribute. During the current chunk processing we are considering both old and current statistics (centres and weights). This is helpful in improving detection rate and reduces false alarms in evolving data streams.

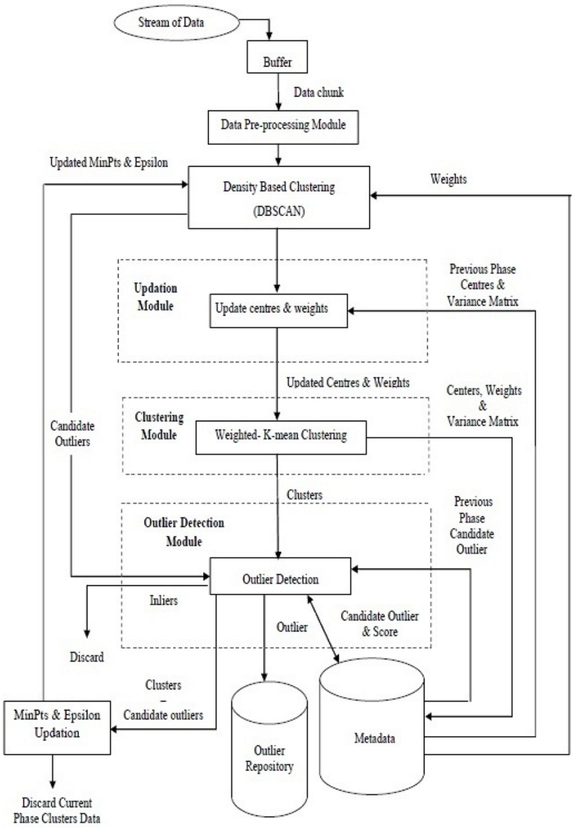
Stream of Data
Buffer
Data chunk
Data Pre-processing Module
Updated MinPts & Epsilon
Density Based Clustering (DBSCAN)
Weights
Updation Module
Update centres & weights
Previous Phase Centres & Variance Matrix
Updated Centres & Weights
Clustering Module
Weighted- K-mean Clustering
Centers, Weights & Variance Matrix
Candidate Outliers
Clusters
Outlier Detection Module
Previous Phase Candidate Outlier
Outlier Detection
Inliers
Discard
Outlier
Candidate Outlier & Score
Clusters – Candidate outliers
MinPts & Epsilon Updation
Outlier Repository
Metadata
Discard Current Phase Clusters Data

Fig. 1. Block Diagram of Proposed Scheme: Unsupervised Outlier Detection in Streaming Data Using Weighted Clustering

E. Clustering Module

In this module data chunk is again clustered using weighted-k-mean [13] clustering method. It is a partitioning clustering algorithm. Updated centres and weights from updation module are used as initial centre and initial weight in clustering. Weighted-k-mean is a k-mean based clustering algorithm which includes iterative weight updation over the clustering process according to equation (3) and (4). In such a manner that intra cluster dissimilarity minimize and inter cluster dissimilarity maximize. Details can be found in [13]. Clusters, weights and variance matrix are the output of this module. Clusters are input to outlier detection module. Cluster centres, weights and variance matrix are saved in metadata repository and old once are discarded.

F. Outlier Detection Module

Outlier detection process comprises three steps. In first step candidate outliers are figure out using the clusters, deviation criteria [12] and threshold and their score is initialised to one. Candidate outliers and their score are saved to metadata repository. In second step previous phase candidate outliers

TABLE I
CHARACTERISTICS OF THE DATA SETS

Data set name	Number of instances	Number of attributes	Number of Normal clusters
Shuttle	58000	9	3
Yeast	1484	8	4
Synthetic dataset	5000	6	4

are tested whether they still satisfy deviation criteria or not. If they satisfy then their score is checked if it is less then L then it is incremented by one and these objects with their score are added to candidate outliers of current phase on metadata repository. If previous phase candidates satisfy deviation criteria and their score is equal to L then these objects are real outlier and stored in outlier repository. If previous phase candidates do not satisfy criteria then those are inliers and discarded. We have checked candidate outliers for consecutive L data chunks before declaring them as outliers or inliers because a cluster may be split over two data chunks or new cluster may be emerging and candidate outlier of current data chunk may become inlier when similar data objects come in next chunks.When new candidate outliers and scores are saved in metadata repository, old once are deleted.

G. MinPts & Epsilon Updation

This module takes the clusters as input from which candidate outlier objects have been removed. These clusters are used according to equation (1) and (2) to update values of MinPts and Epsilon. These equations are based upon the distribution of objects and density of clusters. Now clusters are deleted from memory to empty space for next coming data chunk.

H. Meta Data

Meta data stores Centres of clusters, attribute weights, variance matrix for weight updation, candidate outlier of L phases and their corresponding outlier score, value of L, threshold.

IV. EXPERIMENTAL RESULTS

We have done all implementation in matlab R2010a.We have compared our results with Cluster based OutlieR Miner (CORM) [6] which is a recently proposed outlier detection method for data streams. In next coming subsection we will focus on performance analysis of proposed method in terms of outlier detection rate, false alarm rate, effect of increasing number of irrelevant attributes on detection rate, and effect of increasing percentages of outliers.

A. Data Sets

Experiments are conducted on synthetic as well as real data sets (Table 1). Real data sets were taken from UCI machine learning repository [14].All data sets have numeric attributes.

In Shuttle data set there are 7 classes coded as Rad Flow, Fpv Close, Fpv Open, High, Bypass, Bpv Close Bpv Open. Approximately 80% of original data belongs to class 1. For a better performance analysis we have planted additional data objects (20 % of original data) for class 2 and class 3 and scattered outliers based on the statistical characteristics (min, max, mean and standard deviation) of attributes. Other classes are treated as outlier groups. We have removed first attribute as it is the time but it has be randomised so it is of no use in outlier detection. There are total 10 classes in yeast data. CYT, NUC, MIT, ME3 cover maximum of data instances and other classes covers are small portion of data. From E2, ME1, EXC, VAC AND POX classes we have removed 50% of their data instances to treat them as outlying objects. In our experiments we have not taken sequence name attribute for outlier detection. In synthetic data there are 20 groups of outliers and 50 scattered outliers. All outliers are planted based upon domain knowledge (statistical characteristics like mean, standard deviation, class distribution, type of attributes).

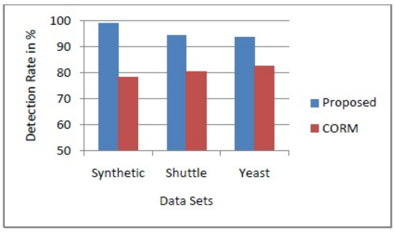

Fig. 2. Outlier Detection Rate

B. Metrics for Measurement

To evaluate the performance of the algorithms we examine two metrics that are outlier detection rate and false alarm rate. Detection rate refers to the ratio between the numbers of correctly detected outliers to the total number of actual outliers. False alarm rate is the ratio between the numbers of normal objects that are misinterpreted as outlier to the total number of alarms. The two metrics are defined in (5) and (6):

$$\text{Detection Rate} = \frac{\text{True Positive}}{\text{True Positive+True Negative}} \tag{5}$$

$$\text{False Alarm Rate} = \frac{\text{False Positive}}{\text{False Positive+False Negative}} \tag{6}$$

C. Outlier Detection Rate & False Alarm Rate

In this section outlier detection rate and false alarm rate of proposed method and CORM is compared on all three data sets.

Fig. 2 shows that on proposed method has much better outlier detection rate than CORM. It is because proposed

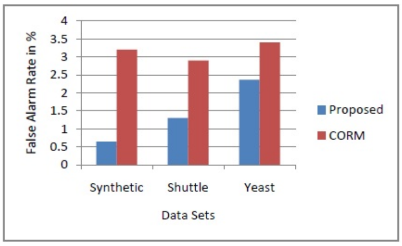

Fig. 3. False Alarm Rate

scheme uses initially DBCAN for detecting possible outlying objects and uses weighted attributes in all computations. It can be observed from Fig. 3 that in terms of false alarm rate proposed scheme performs better than CORM.

D. Effect of Increasing Number of Irrelevant Attributes on Detection Rate

For analysing the effect of increasing number of irrelevant attribute we have used synthetic data set and added irrelevant attribute (uniformly distributed) [15] one after another.

It can be observed from Fig. 4 that outlier detection rate of proposed method is much consistent with increasing number of noisy attributes as compare to CORM. This because proposed method automatically assigns weights to attributes based on their respective significance. Smaller weights are given to noisy attributes and larger weights to more relevant attributes. Smaller weights reduce the effect of noisy attributes. CORM gives equal importance to all attributes and noisy attributes hinder in outlier detection.

E. Effect of Increasing % of Outliers Groups and Scattered Outliers

In this section we compare the outlier detection rate of proposed method with CORM when the percentage of outliers in the form of outlier groups and scattered (individual) outliers is varied. For this experiment we have used synthetic data set and artificial outliers are planted in increasing percentage in the form of outlier groups and scattered outliers. This percentage of outliers is taken corresponding to original data set size. We have taken synthetic data set because it is easy to control all its all parameters of it.

From Fig. 5 it is clear that outlier detection rate of CORM is decreasing very fast with increasing percentage of outliers as compare to proposed method. Proposed methods outlier detection rate is more consistent. It is because proposed method treats small clusters as outlier groups instead of normal clusters following the criteria of [5] for large and small size of clusters. There is a fall in detection rate of proposed method when size of outlier group is much larger than other clusters size. Also we can say that these objects no more have anomalous behavior as unsupervised clustering based outlier detection approaches assume that outlier are a small fraction of whole data. But for finding accuracy we have still taken them as outliers. So this is an obvious decrement in detection rate.

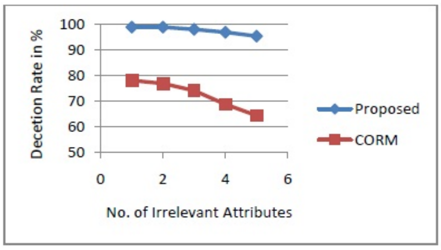

Fig. 4. Effect of Increasing Number of Irrelevant Attributes on Outlier Detection Rate

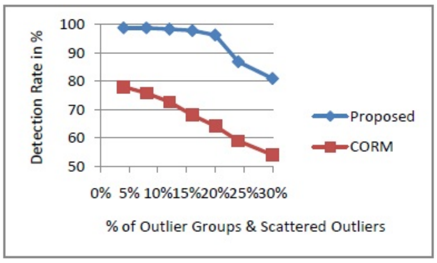

Fig. 5. Effect of Increasing % of Outliers Groups and Scattered Outliers on Detection Rate

V. Conclusion

In this paper, we have presented a clustering based outlier detection scheme for streaming data. This scheme has applied both density based (DBSCAN) and partitioning (weighted-k-mean) clustering for detection of individual as well as group of outliers. Weighted-k-mean clustering is also used to assigns weights to attributes depending upon their respective relevance in clustering. To face the challenges of data stream processing our proposed scheme is incremental and dynamic in nature. We have processed streaming data in the form of data chunks and candidate outliers are checked over multiple consecutive data chunks before declaring them as outliers or inliers. After processing a data chunk only necessary statistics of chunk are kept and chunk is discarded to free up memory for next chunk. Weights and centers are adaptive to concept evolution.

Experimental results show that the proposed method gives higher outlier detection rate and lower false alarm rate than CORM. The performance of proposed scheme is very consistent with increasing number of noisy attributes as compare to CORM. The proposed method has performed much better than CORM with increasing percentages of outliers.

In future we will extend our method for categorical and mixed data types.

References

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, J. Kacprzyk and L. C. Jain, Eds. Morgan Kaufmann, 2006, vol. 54, no. Second Edition.

[2] Yogita and D. Toshniwal, "A framework for outlier detection in evolving data streams by weighting attributes in clustering," in *Proceedings of the 2nd International Conference on Communication Computing and Security*, India, 2012.

[3] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 427–438.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 93–104.

[5] Z. He, X. Xu, and S. Deng, "Discovering cluster based local outliers," *Pattern Recognition Letters*, vol. 2003, pp. 9–10, 2003.

[6] M. Elahi, K. Li, W. Nisar, X. Lv, and H. Wang, "Efficient clustering-based outlier detection algorithm for dynamic data stream," in *Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery - Volume 05*, ser. FSKD '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 298–304.

[7] F. Angiulli and F. Fassetti, "Detecting distance-based outliers in streams of data," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, ser. CIKM '07. New York, NY, USA: ACM, 2007, pp. 811–820.

[8] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *Proceedings of the 32nd international conference on Very large data bases*, ser. VLDB '06. VLDB Endowment, 2006, pp. 187–198.

[9] M. S. Sadik and L. Gruenwald, *DBOD-DS : Distance Based Outlier Detection for Data Streams*. Springer, 2011, vol. 6261, p. 122136.

[10] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," *IEEE Trans. on Knowl. and Data Eng.*, vol. 18, no. 2, pp. 145–160, Feb. 2006.

[11] L. Duan, L. Xu, Y. Liu, and J. Lee, "Cluster-based outlier detection," *Annals of Operations Research*, vol. 168, pp. 151–168, 2009.

[12] M. B. Al-Zoubi, "An effective clustering-based approach for outlier detection," *European Journal of Scientific Research*, vol. 28, pp. 310–316, 2009.

[13] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, May 2005.

[14] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[15] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 3:1–3:39, Mar. 2012.