ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

High Dimensional Anomaly Detection algorithms for Mining High Speed Data Streams in Massive Online Analysis (MOA) Framework

Konda Sreenu
Assistant Professor in Computer Science and Engineering Department,
Sir C R Reddy College of Engineering, Eluru, +919490970060
sreenukcupid@gmail.com

Abstract- This paper illustrates the problems encountered in detecting and fixing anomalies in high-dimensional data Streams. In high dimensional data streaming the data arrives at a fast phase in multidimensional space, and interrelation between different data sets is complex. Because of these reasons finding meaningful anomalies becomes more complex and non-evident. Moreover, in high Dimensional Streaming the data become sparse, it makes all points to look very similar. Also, due to fast phase streaming, the full datasets required for analysis will not available in short span of time. Hence, analyses of high dimensional data streams are totally different from that of normal data Streams, and so it requires different approaches. Network Robustness, intrusion detection, etc. are some of the current applications used in the analysis of high dimensional data streaming. Recently, many new algorithms have been developed to find and rectify the anomalies. Massive Online Analysis framework is a distance based cluster an algorithm that can be used for mining high dimensional data sets. This algorithm is used to determine the anomalies using Simple Continuous Outlier Detection (Simple COD) algorithm and Micro cluster Continuous Detection (MCOD) algorithm. Both these algorithms are used to compare data sets of different size and type. A comparative a study of both these algorithms is done and the results are presented herein.

Keywords -Data mining, High speed data Streams, Data Streams, High Dimensional anomaly detection, Outlier Detection, MOA, Simple COD, and MCOD.

I. INTRODUCTION

An anomaly is a data point which substantially varies from the remaining data. It is also referred as outer, Abnormality, discordant, deviant, etc., in the data mining. In most of applications, the data will be created by one Or more data generating processes. The data generation An activity conducted in an unusual way, results in creation Of anomaly. Therefore, an anomaly frequently keeps the Useful information about defective characteristics of the Systems and entities, which affect the data generation Process. Analysis of such unusual property furnishes Useful application-specific insights.

A. The basic anomaly models

An anomaly in data streaming can occur due to several Factors, including type and size of data. Thus, it is highly important to have meaningful and easily interpretable Anomaly detection models. Such models can be used to determine why a specific the data point is an outlier with respect to the rest of the data Sets. And to diagnose a different theoretical scenario that could lead to anomaly. Different functionality off Anomaly models are required to study different features of data structures. For example, to see the data Transformation from animals to normal data visibly a color change associated with such transformation would be easily visible seeing it. Therefore, it is very difficult, but important to choose a specific model for anomaly Analysis.

High-dimensional data streaming methods furnish an Interesting direction for meaningful interpretation of Anomaly analysis results. The output of the algorithms Provide a specific group of attributes along with a data point having the anomaly. This kind of interpretability is very helpfully, when a small number of attributes need to be selected from a large number of possibilities in anomaly Analysis.

B. Classification

In traditional batch learning the problem of limited data is overcome by analyzing and averaging multiple models produced with different random arrangements of training and test data. In the stream setting the problem of (effectively) unlimited data poses different challenges. One solution involves taking snapshots at different times during the induction of a model to see how much the model improves.

The evaluation procedure of a learning algorithm determines which examples are used for training the algorithm, and which are used to test the model output by the algorithm. When considering what procedure to use in the data stream setting, one of the unique concerns is how to build a picture of accuracy over time. Two main approaches arise:

• Holdout: When traditional batch learning reaches a scale where cross-validation is too time consuming, it is often accepted to instead measure performance on a single holdout set. This is most useful when the division between train and test sets has been pre-defined, so that results from different studies can be directly compared. • Interleaved Test-Then-Train or Prequential: Each individual example can be used to test the model before it is used for training, and from this the accuracy can be incrementally updated. When

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

intentionally performed in this order, the model is always being tested on examples it has not seen. This scheme has the advantage that no holdout set is needed for testing, making maximum use of the available data. It also ensures a smooth plot of accuracy over time, as each individual example will become increasingly less significant to the overall average (Gama et al., 2009).

Considering data streams as data generated from pure distributions, MOA models a concept drift event as a weighted combination of two pure distributions that characterizes the target concepts before and after the drift. Within the framework, it is possible to define the probability that instances of the stream belong to the new concept after the drift. It uses the sigmoid function, as an elegant and practical solution (Bifet et al., 2009a, b).

MOA contains the data generators most commonly found in the literature. MOA streams can be built using generators, reading ARFF files, joining several streams, or filtering streams. They allow for the simulation of a potentially infinite sequence of data. The following generators are currently available: Random Tree Generator, SEA Concepts Generator, STAGGER Concepts Generator, Rotating Hyper plane, Random RBF Generator, LED Generator, Waveform Generator, and Function Generator. The implemented classifier methods currently include: Naive Bayes, Decision Stump, Hoeffding Tree, Hoeffding Option Tree (Pfahringer et al., 2008), Bagging, Boosting, Bagging using ADWIN, and Bagging using Adaptive-Size Hoeffding Trees (Bifet et al., 2009b).

C. Clustering

MOA contains also an experimental framework for clustering data streams, which allows comparing different approaches on individual (real world) settings and which makes it easy for researchers to run and build experimental data stream benchmarks. The features of MOA for stream clustering are:

- Data generators for evolving data streams (including events such as novelty, merge, etc. (Spiliopoulou et al., 2006)),
- An extensible set of stream clustering algorithms,
- Evaluation measures for stream clustering,
- Visualization tools for analyzing results and comparing different settings.

For stream clustering us added new data generators that support the simulation of cluster evolution events such as merging or disappearing of clusters (Spiliopoulou et al., 2006). Currently MOA contains several stream clustering methods such as StreamKM++ (Ackermann et al., 2010), CluStream (Aggarwal et al., 2003), ClusTree (P. et al., 2010), Den-Stream (Cao et al., 2006), D-Stream (Tu and Chen, 2009) and CobWeb (Fisher, 1987). Moreover, MOA contains measures for analyzing the performance of the clustering models generated including measures commonly used in the literature as well as novel evaluation measures to compare and evaluate both online and offline components. The available measures evaluate both the correct assignment of examples (Chen, 2009) and the compactness of the resulting clustering.

[5] discussed about a method, Wireless sensor networks utilize large numbers of wireless sensor nodes to collect information from their sensing terrain. Wireless sensor nodes are battery-powered devices. Energy saving is always crucial to the lifetime of a wireless sensor network. Recently, many algorithms are proposed to tackle the energy saving problem in wireless sensor networks. There are strong needs to develop wireless sensor networks algorithms with optimization priorities biased to aspects besides energy saving. In this project, a delay-aware data collection network structure for wireless sensor networks is proposed based on Multi hop Cluster Network. The objective of the proposed network structure is to determine delays in the data collection processes. The path with minimized delay through which the data can be transmitted from source to destination is also determined. AODV protocol is used to route the data packets from the source to destination.

Besides providing an evaluation framework, the second key objective is the extensibility of the benchmark suite regarding the set of implemented algorithms as well as the available data feeds and evaluation measures.

The visualization component allows visualizing the stream as well as the clustering results, choosing dimensions for multi dimensional settings, and comparing experiments with different settings in parallel. For example, a screen shot of our visualization tab. For this screen shot two different settings of the CluStream algorithm (Aggarwal et al., 2003) were compared on the same stream setting (including merge/split events every 50000 examples) and five measures were chosen for online evaluation (CMD, F1, Precision, Recall and SSQ). The upper part of the GUI offers options to pause and resume the stream, adjust the visualization speed, choose the dimensions for x and y as well as the components to be displayed (points, micro- and macro clustering and ground truth). The lower part of the GUI displays the measured values for both settings as numbers (left side, including mean values) and the currently selected measure as a plot over the arrived examples (right, F1 measure in this example). For the given setting one can see a clear drop in the performance after the split event at roughly 160000 examples (event details are shown when choosing the corresponding vertical line in the plot). While this holds for both settings, the left configuration (red, Clustered with 100 micro clusters) is constantly outperformed by the right configuration (blue, Clustered with 20 micro clusters).

D. Website, Tutorials, and Documentation

The Http://moa.cs.waikato.ac.nz/ website includes a tutorial, an API reference, a user manual, and a manual about mining data streams. Several examples of how the software can be used are available. For example, a nontrivial example of the EvaluateInterleaved TestThenTrain task creating a comma separated values file, training the HoeffdingTree classifier on the Waveform Generator data, training and testing on a total of 100 million examples, and testing every one million examples, is encapsulated by the following command line: java- cp.: moa. jar: weka.Jarjavaagent: sizeofag.jar

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

oa.DoTask \ "EvaluateInterleavedTestThenTrain -l HoeffdingTree \ -s generators.WaveformGenerator\ -i 100000000 -f 1000000" > htresult.csv

MOA is easy to use and extend. A simple approach to writing a new classifier is to extend moa.classifiers. AbstractClassifier, which will take care of certain details to ease the task. Although the current focus in MOA is on classification, we plan to extend the framework to include data stream clustering, regression, and frequent pattern learning (Bifet, 2010).

E. Some of the Practical examples

Credit Card fraud detection systems and Sensor Events: Nowadays getting sensitive information such as credit card number, CCV number, etc. is very prevalent. These data theft generally lead different patterns of unauthorized use of Credit cards like different geographical locations. Such Patterns can be used as input to find anomalies in credit Card transactions**.** Sensor events are frequently used to Trace various types of criminals using many real time Applications including location of using fast changes in Various environmental factors like location is used as Fundamental patterns. Finding event is one of the basic Applications in the field of sensor networks.

Intrusion Detection Systems: The networked or host based Systems have various types of data which are Collected from sources such as network traffic, operating System call, etc. This system will show the unusual Activity because of malice. The detection of such Abnormality is referred as intrusion detection system.

Implementation of the Law: Anomaly detection system is used in a number of applications to implement the law, Such as finding fraud in financial transactions, trade, Insurance claims, etc. The crime is detected by Monitoring the unusual patterns (data anomaly) over time In multiple activities, the occurrence of such unusual Patterns in the data can only be generated by criminal Activity.

Earth Sciences: The data about weather, climate Changes are collected from various sources. Anomalies With such data furnish substantial understandings about Hidden human and environmental turns.

Medical Diagnosis system: In a medical diagnosis System, medical devices such as PET scans or ECG Time-series, MRI, etc. produces different patterns. Unusual patterns in such data are generally evident Disease condition.

II. RELATED WORK

Many authors around the world already rendered numbers of literary study based on anomaly detection. Many of them are density based, partition based or distance based techniques, Sliding window based anomaly detection and many more new ideas are being developed in order to find solutions for the problems in high speed streaming data.

[1] Discuss a hybrid approach over higher dimensional streamed data which is extremely useful in real time work. In this approach, high streamed data is divided into a number of chunks at a particular time over every data chunk that comes across during that time. The hybrid method comprises a cluster based and distance based approaches. The data stream is divided into many clusters of data points. In distance based method, the threshold for each cluster, pruning out points lying outside the radius determined. In this hybrid approach, the data points which lie outside the threshold will be used as a final candidate outlier.

[2] Discuss about the link anomaly detection technique, which finds anomaly in the data stream by using clustering method. Probability model and dynamic link optimization algorithm are used for detecting anomalous. Through investigation, it is claimed that the accuracy of this method is more than many other methods.

[3] Presented a clustering based method to capture anomaly. Here K-means clustering algorithm is applied to split the data set into clusters. In the result the points, which exist near the centroid of the cluster are not likely anomaly and can prune out such points from each cluster. Based on the outlier score found, they declare the top n points with the highest score as anomaly factor. The authors performed the experiments using high dimensional data sets. This method was found to be produce better anomaly detection, better than the existing method.

A. High speed data stream vs. Anomaly

Time-series data streams contain a group of values which are bringing into existence by continuous measurement around the time. Therefore, the values in successive time-stamps do not alter very significantly, or changes in an easy way. In such cases, fast change in the lying data records, can be considered anomalous events. Therefore the discovery of anomalous points in time series stream is usually closely associated to the problem of anomalous event detection, in the form of either environmental or related to time-stamp. Generally such events are created by a fast change in the lying system, and may be of significant benefit to an analyst. For example, let us assume the following time-series streams of values, 4, 3, 4, 3, 4, 88, 87, 86, 88, 90, 96, 4, 85, 92, 87, 92, 89

The time-series data stream is explained in Figure 1. It is proof that there is a fast changes in the data value at time-stamp 6 from 4 to 88. This equals to an anomaly. Afterwards, the data changes at this value, and this becomes the new pattern. At time-stamp 12, the data value again brings downs to 3 then it is considered an anomaly because of the fast change in the successive data values. Thus, it is crucial to understand that in this case, handling the data values independence of one another is not helpful for anomaly

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

detection, because the data values are highly affected by the next values of the data points. Thus, the problem of anomaly detection in time series data stream is highly related to the problem of change detection, because the usual models of data values are extremely ordered by adjacency in temporal arranging. When completely new data values are found, they are denoted as to as novelties though anomaly detection is applicable to any form of fast change, rather than only originalities, which are a particular kind of anomalies. It should be expressed that change analysis and anomaly detection in temporary data are very closely related areas, but not necessarily same.

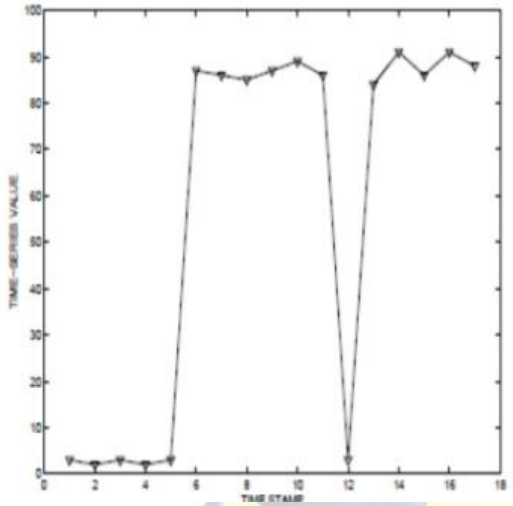


Figure 1. Time Series data.

ii) Second, the algorithm is selected; iii) finally evaluation or measure is chosen. After this setup, define the size of the sliding window, streaming data mining to run and produce the result. All the above three phases are extensible. Thus MOA is used to handle huge, possibly countless, developing data streams. MOA mainly allows the evaluation of data stream learning algorithms on massive streams under unambiguous memory limits. The anomaly mining algorithm set up mostly included the following steps; i) Select the stream ii) Select algorithm1(Simple COD) iii) Select algorithm2(MCOD). The visualization window is mainly displays the performance of the selected algorithms for a fixed number of occurrences.

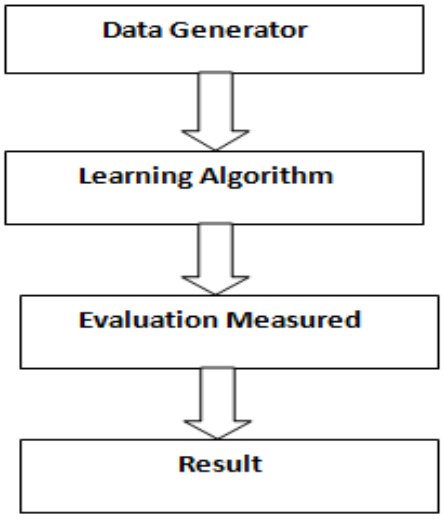


Figure 2. MOA Framework

B. High dimensional data

Since high dimensional data streams, changes in the aggregate assignment of the streaming data may represent to unusual events. For example, network intrusion factors may cause aggregate change points in a network stream. On the other hand, single point novelties may or may not represent to aggregate change points. The second case is similar to high dimensional anomaly detection with an efficiency restriction for the streaming assumption.

III. SYSTEMS AND MODEL

A. A Massive Online Analysis (MOA) Frameworks

MOA is open-source framework software that allows building and running experiments of machine learning or data mining on evolving data streams. It includes a set of learners and stream generators that can be used from the Graphical User Interface (GUI), the command-line, and the Java API.

MOA (Massive Online Analysis) is an open source tool that builds the work in WEKA2. Its main aim is to provide a simple extensible for enforce, evaluate and appraise classification, anomaly detection for streaming data. The MOA system architecture involves three phases for any data mining task: i) first, a data stream is selected and assembled;

B. Data Generator

Random RBF Generator This generator was discovered to offer an alternate difficult concept type that is not free from ambiguity to estimate with a decision tree model. The RBF (Radial Basis Function) generator works as follows: A fixed number of random center objects are generated. Each center has a class label, a random position, a single standard deviation and weight. New examples are given by selecting a center at random, taking weights into condition so that centers with more weights are to be chosen. A random direction is chosen to offset the attribute values from the central point. The length of the movement is randomly drawn from a Gaussian distribution with standard deviation fixed by the chosen central object. The chosen central object also fixes the class label of the example. This effectively creates a normally distributed hyper sphere of examples surrounding each central point with changing densities. Only numeric attributes are generated. Drift is introduced by moving the central object with constant speed. This speed is initialized by a drift parameter.

LED Generator This data source develops from the CART book an implementation in C was donated to the UCI machine learning repository by David Aha. The goal is to predict the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

digit displayed on a seven-segment LED display, where each attribute has a 10% chance of being inverted. It has an optimal Bays classification rate of 73.9%. The particular configuration of the generator used for experiments.

Waveform Generator The goal of the task is to distinguish between 3 dissimilar classes of waveform, each of which has generated from a combination of 2 or 3 base waves. The optimal Bays classification rate is known to be 85%.

Functions Generator It was a general source of data for former work on scaling up decision tree learners. The generator produces a stream comprising 9 attributes, 6 numeric and 3 categorical. For example these attributes identify hypothetical loan applications. There are 10 functions are fixed for generating binary class labels from the attributes, these determine whether the loan should be approved or not.

C. *Learning algorithms used in the anomaly detection*

Continuous anomaly detection is a special class of steam data mining. Typically, stream data mining algorithms assume that each object is inspected at most once. However, in continuous outlier detection we need to be capable of reporting, at each time point, the anomalies among all the objects in the current sliding window. This means that we need to continuously visit each object that has not expired (either directly or indirectly) rather than visiting it only once. The reason is that an object may change its anomalies status during its lifetime. This characteristic changes the need for high time and space efficiency.

The criteria according to which an object is classified as an outlier may vary across different techniques. In this work, we focus on distance-based anomalies, which catch a broad range of assumptions: Given two parameters, R >= 0 and k>= 0, a distance-based anomaly is every object that has less than k adjoin in distance at most equal to R. Moreover, we are mostly interested in exact algorithms. In the continuation, we will briefly describe 4 distance-based continuous anomaly algorithms that we are also going to demonstrate.

D. *Evaluation measured*

In conventional batch learning the problem of limited data is get over by examining and averaging many models produced with different random arrangements of training and test data. In the stream setting the problem of unlimited data arise different challenges. One result involves taking exposure at different times during the induction of a model to see how much the model improves. The evaluation procedure of a learning algorithm decides which examples are used for training the algorithm, and which are used to test the model output by an algorithm. When look at what procedure to use in the data stream setting, one of the unequaled concerns is how to build an image of accuracy over time. Therefore there are two main approaches are followed. They are as follows.

Holdup: When conventional batch learning reaches as an order where cross validation is time taken process, it is often accepted to instead measure performance on a single holdout set. This is most useful when the division between train and test sets has been pre-defined, so that results from different studies can be directly compared.

Test-Then-Train: Each example can be used to test the model before it is used for training, and from this the accuracy can be additive updated. When deliberately performed in this order, the model is ever being tested on examples it has not seen. This scheme has the advantage that no holdup set is needed for testing, making maximum use of the available data. It also assures a smooth plot of accuracy over time, as each and every example will become increasingly less significant to the overall average.

E. *Comparisons of an algorithms*

To mine the anomalies, there are used simple Continuous Outlier Detection (Simple COD) algorithms and Micro Cluster based Continuous Detection (MCOD) algorithm. The increased efficiency of COD algorithm computes the next point due to departure of an object. The object may become an anomaly and visits an object only at that time point. MCOD algorithm develops on top of COD and uses the same event line. Its specifiable characteristics are that it justifies the need to measure queries range for every new object. In huge regions data, MCOD shows the best performance. The COD and MCOD algorithms are combined for development of time and space complexities.

IV. EXPERIMENTAL RESULTS

The view of continuous outlier detection algorithms are performed using MOA's visualization functionality. This concentrate on 2 different faces of continuous anomaly detection; i) the search of the anomaly and ii) the comparison of operations among the implemented algorithms. The search will be based on both synthetic as well as on real-world data sets with different characteristics. For example the Forest Cover dataset from the UCI KDD Archive

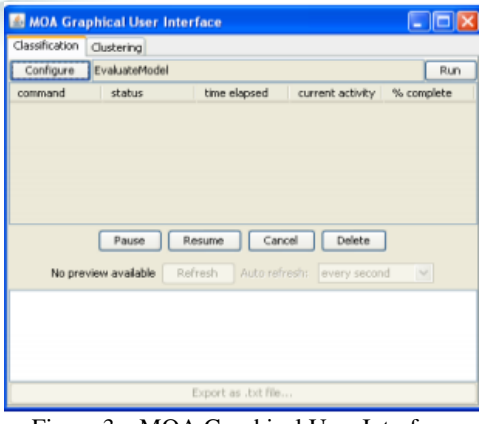


Figure.3a. MOA Graphical User Interface

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

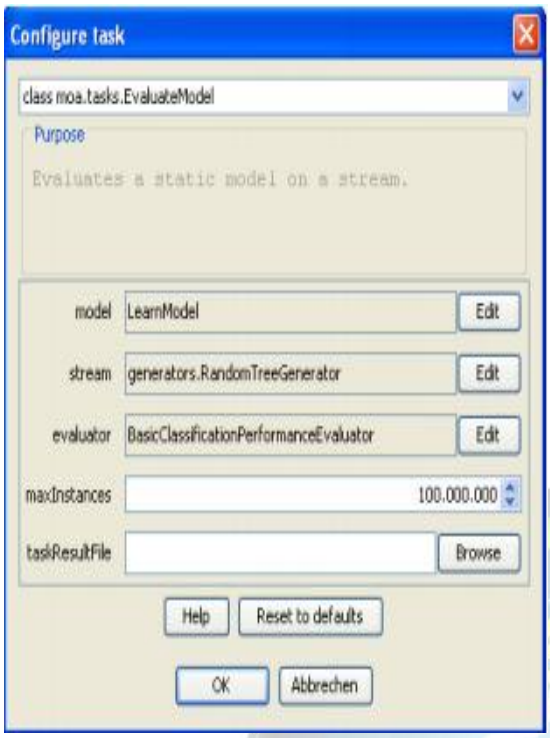


Figure.3b. MOA Graphical User Interface

Figure 3a, 2b shows the MOA graphical user interfaces and a command line interface is also available. This GUI helps us to select the algorithms

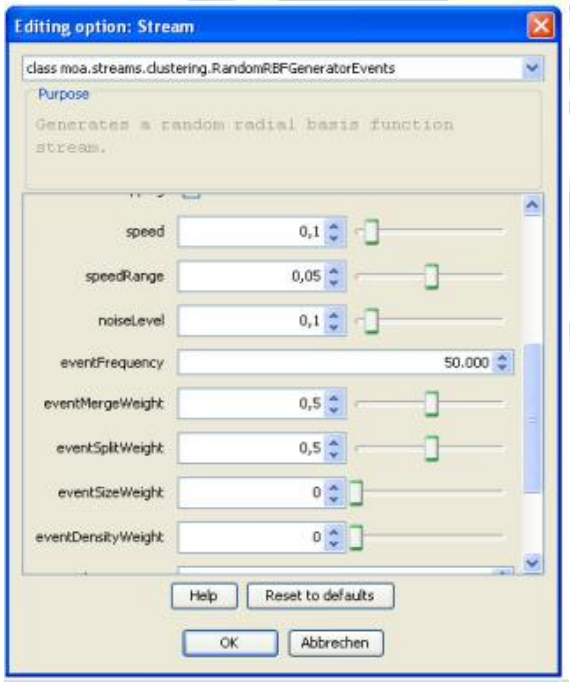


Figure4a. Option dialog for the RBF data generator

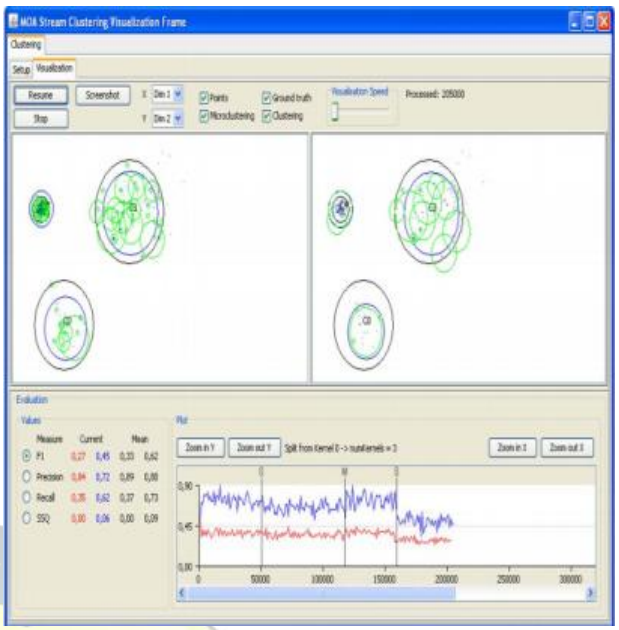


Figure4b. Option dialog for the RBF data generator

Figure 4a, 4b shows option dialog for the RBF data generator (by storing and loading settings benchmark streaming data sets can be shared for repeatability and comparison) visualization tab of the clustering MOA graphical user interface.

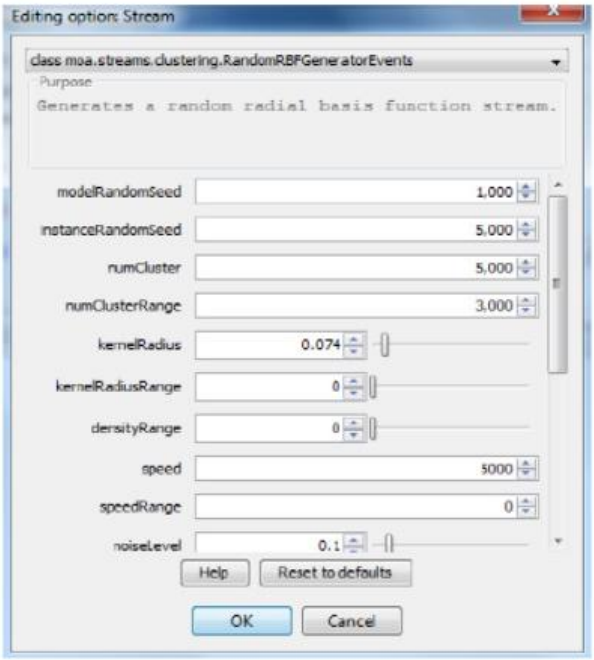


Figure 5. Data Generator

Figure 5 show the MOA data generator and which is used to set the data values for the algorithms.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

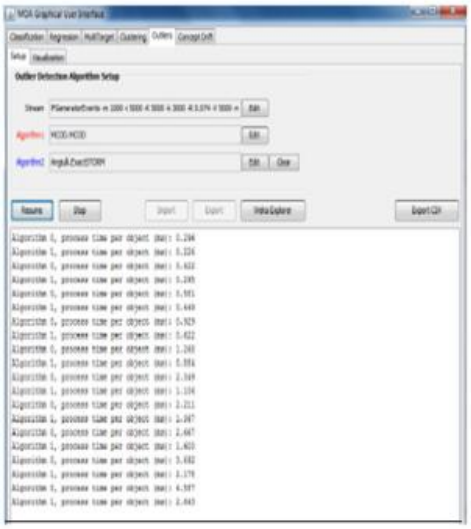


Figure 6. Execution of Time-stamp

Figure 6 show that the time-series data stream which was explained in Figure 1. It is proof that there is a fast changes in the data value at time-stamp 3 to 4 and 19. This equals to an anomaly. Afterwards, the data changes at this value, and this becomes the new pattern. At timestamp 6, the data value again brings downs then it is considered an anomaly because of the fast change in the successive data values.

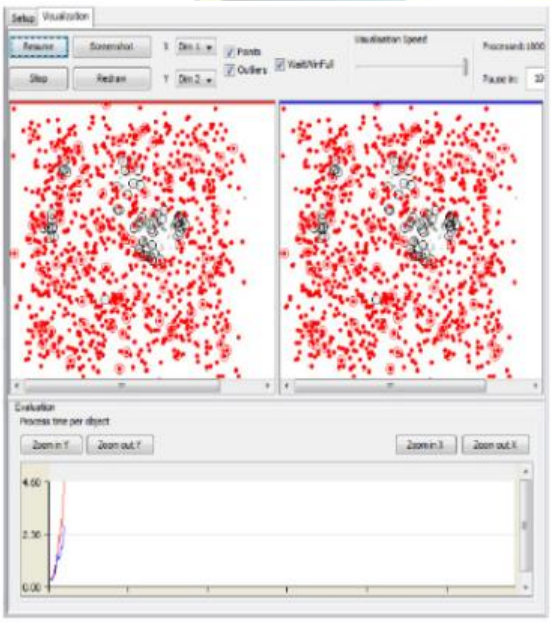


Figure 7. Visualization tab of anomaly detections

Figure 7 show the research involves the exploration of data streams with respect to anomalies and their development. In this part, we will demonstrate visually the way objects change their status during their lifetime inside the sliding window. An example is illustrated in Fig. 6, where in a new object is inserted and become an inliers; the darkest the color, the more recent the point. In the same object becomes an anomaly due to object departure in its R-neighborhood and this change is denoted by a red circle. Finally, the object becomes again inliers due to the arrival of new objects in its Neighborhoods; this is denoted by a black circle. Evidently, by setting different values for the parameters k and R the anomalies change accordingly, and we are able to spot the timestamps that these changes took place.

V. CONCLUSION

Our goal is to build an experimental framework for anomaly detection on data streams similar to the WEKA framework. Our stream learning framework furnishes a set of data generators, algorithms and evaluation measures. The user can benefit from this by comparing several algorithms in real world assumptions and choosing the best suitable solution. This framework allows the creation of standard streaming data sets through stored, shared and repeatable settings for the data flows. The sources are publicly available and are released under the GNU GPL license. Although the current focus in MOA is on classification and clustering, we plan to extend the framework to include regression.

REFERENCES

[1] M. R. Ackermann, C. Lammersen, M. M¨artens, C. Raupach, C. Sohler, and K. Swierkot. Stream KM++: A clustering algorithm for data streams. In SIAM ALENEX, 2010.

[2] C. C. Aggarwal, J. Han, J .Wang, and P. S. Yu. A framework for clustering evolving data streams. In VLDB, pages 81–92, 2003

[3] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos. Continuous monitoring of distance-based outliers over data streams. In ICDE, pages 135–146, 2011

[4] F. Angiulli and F. Fassetti. "Distance-based outlier queries in data streams: the novel task and algorithms". Data Mining and Knowledge Discovery, 20(2):290–324, 2010.

[4] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos. Continuous monitoring of distance-based outliers over data streams. In ICDE, pages 135–146, 2011

[5] Christo Ananth, T.Rashmi Anns, R.K.Shunmuga Priya, K.Mala, “Delay-Aware Data Collection Network Structure For WSN”, International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Special Issue 2 - November 2015, pp.17-21

[6] Bifet, A. Holmes,G, Pfahringer,B., Kirkby,R., and Gavaldà, R. "New ensemble methods for evolving data streams," Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.,139- 148,2009, ACM.

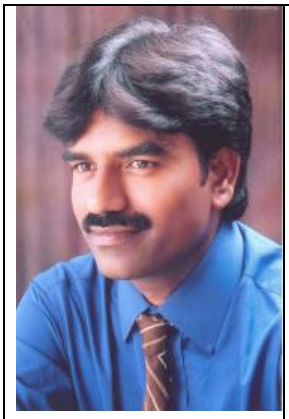
[7] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. MOA: Massive Online Analysis (MOA) http://sourceforge.net/projects/moa-datastream/. Journal of Machine Learning Research (JMLR), 2010.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

[8] J. Gama, R. Sebasti˜ao, and P. P. Rodrigues. Issues in evaluation of stream learning algorithms. In 15th ACM SIGKDD, 2009.
[9] P. Kranen, I. Assent, C. Baldauf, and T. Seidl. Self-adaptive anytime stream clustering. In IEEE ICDM, pages 249–258, 2009.
[10] M. J. Song and L. Zhang. Comparison of cluster representations from partial second- to full fourth-order crosses moments for data stream clustering. In ICDM, pages 560–569, 2008
[11] L. Tu and Y. Chen. Stream data clustering based on grid density and attraction. ACM Trans. Knowl. Discov. Data, 3(3):1–27, 2009.
[12] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. MONIC: modeling and monitoring cluster transitions. In ACM KDD, pages 706–711, 2006.
[13] Marcel R. Ackermann, Christiane Lammersen, Marcus M¨artens, Christoph Raupach, Christian Sohler, and Kamil Swierkot. StreamKM++: A clustering algorithm for data streams. In SIAM ALENEX, 2010.
[14] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In VLDB, pages 81–92, 2003.
[15] Kranen P., Assent I., Baldauf C., and Seidl T. The ClusTree: Indexing micro-clusters for anytime stream mining. In Knowledge and Information Systems Journal (KAIS), 2010.
[16] Feng Cao, Martin Ester, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In SDM, 2006.
[17] J. Chen. Adapting the right measures for k-means clustering. In ACM KDD, pages 877–884, 2009
[18] Albert Bifet. Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams. IOS Press, 2010.
[19] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, and Ricard Gavalda. Improving adaptive bagging ` methods for evolving data streams. In First Asian Conference on Machine Learning, ACML 2009, 2009a.
[20] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Richard Kirkby, and Ricard Gavalda. New en- ` semble methods for evolving data streams. In 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009b.
[21] Bernhard Pfahringer, Geoff Holmes, and Richard Kirkby. Handling numeric attributes in hoeffding trees. In PAKDD Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 296–307, 2008
[22] Joao Gama, Raquel Sebasti˜ao, and Pedro Pereira Rodrigues. Issues in evaluation of stream learning algorithms. In 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.

BIBLIOGRAPHY OF AUTHORS



Sri. Konda Sreenu, working as Assistant Professor in Computer Science and Engineering Department, Sir C R Reddy College of Engineering, Eluru. He has an experience of 11 years in teaching. He is currently pursuing PhD from Acharya Nagarjuna University, Guntur. He has completed his M.Tech from SIT, JNTUH, and Hyderabad. He has completed his B.Tech from Sri Vasavi Engineering College, JNTUK, and Tadepalli gudem, Andhra Pradesh..