TIME-VARIANT OUTLIER DETECTION METHOD ON GEOSENSOR NETWORKS

Dong Phil Kim, Gyeong Min I, Dong Gyu Lee, Keun Ho Ryu

Database/Bioinformatics Laboratory, Chungbuk National University, Korea
{lomego, min9709, dglee, khryu}@dblab.chungbuk.ac.kr

ABSTRACT: Existing Outlier detections have been widely studied in geosensor networks. Recently, machine learning and data mining have been applied the outlier detection method to build a model that distinguishes outliers based on anchored criterion. However, it is difficult for the existing methods to detect outliers against incoming time-variant data, because outlier detection needs to monitor incoming data and classify irregular attacks. Therefore, in order to solve the problem, we propose a time-variant outlier detection using 2-dimensional grid method based on unanchored criterion. In the paper, outliers using geosensor data was performed to classify efficiently. The proposed method can be utilized applications such as network intrusion detection, stock market analysis, and error data detection in bank account.

KEY WORDS: Outlier detection, Sensor network, Grid, Clustering, Classification

1. INTRODUCTION

Data streams that have happened from development of network and hardware are unbounded data such as communication data, bank account, and stock market analysis. The characteristics of a data stream are continuous, high-capacity, enabling only sequential access without database, and enabling only one reading. Classification researches to process data stream effectively on these characteristics have studied now [1].

One-pass algorithm [6] on data streams usually builds a classifier according to data flow and does not allow the classifier that is varied on individual time. The goal of researches that concern variation of data is to support newly a classifier evolution in real-time. Accuracy of such model is less than approaches of Sliding Window [5, 6]. However, models of data flow may be build a classifier concurrently evolved on individual time and test them [2]. Dataset is consisted of 2 kind data as training data and test data. Training data are labelled and test data is not labelled.

These researches do not support variation of data streams on individual time after building a classifier. A data stream is event data such as outlier detection, and communication error. If a classifier does not support variations of data stream on individual time, it is difficult to apply event data to the classifier in real-time. Therefore, time-variant outlier detection varied on individual time is required for solving above problems.

In this paper, we present a time-variant outlier detection that classifies a data stream in real-time according to time attribute. If training dataset comes, it indicates coordinates as relationship between time axis(X axis) and value axis(Y axis). A cell becomes one cluster on 2D coordinates. When individual micro-cluster is on closed Euclidean distance, proposed detection method adjusts whether a data stream is within range defined by users or not. Such range is a cell on 2D grid. Clusters that are within adjacent range are merged to one cluster. A classifier is built of Incoming labelled data stream (training data). If incoming data stream is unlabeled, it is a test data. The classifier is utilized to detect network outlier, transaction error, and stock market analysis.

In the remainder of the paper, we not only explain existing researches that is studied outlier detection, but also describe proposed time-variant outlier detection. Finally, we make a conclusion with future work.

2. RELATED WORK

Clustering technique has studied on data stream environments. BIRCH is to cluster unbounded data stream using minimum memory [7]. It builds CF-tree for such clustering. It performs using information summarized in limited memory. If CF-tree is not built in memory due to a volume of data, it is built of filtered data according to the threshold. CF-tree can cluster efficiently in limited memory [2, 7]. There are also CURE, ROCK, CLARANS in previous work. They are sampling methods and have low accuracy. Therefore, it is not stable on data stream environments. DBSCAN[8] is a method based on density. To compute density, it needs whole data. Therefore, it is difficult to apply time-variant outlier detection method on data stream environments because it stores all data on data stream environments.

LOCAL SEARCH is clustering technique on data stream environments [4]. The technique stores dataset and clusters assigning the weight according to the number of stored data in particular cluster. It does not support to cluster in real-time according to temporal variation and only considers efficient clustering. CluStream has been studied [3]. It stores CF-tree every regular time and compares current CF-tree with already stored CF-tree. This study focuses on difference between current and previous CF-tree. If a time interval increases, the accuracy of clustering decreases.

There is a study of evolving model as On Demand Classification [2]. This technique clusters a data stream as training data and then adjusts whether test data contain within a cluster. A method for clustering is K-means

algorithm [2, 3]. CF-tree includes summary information of individual cluster. Clusters containing the summary information are assigned unique identification number. When test data is entered, it adjusts whether the data is within a cluster or not. When training data and test data is incoming concurrently, it has a disadvantage that K-means performs slowly on data stream environment.

Also, many researchers get along with many work of outlier detection. Statistical methods usually handle detection of abnormal data [12]. This method detects very true using previous empirical data. Detection method observes events of normal cluster and generates profile of individual event. Profiles created are observed at regular time for detecting abnormality. Strength of statistical method will apply in various areas of statistical study very well. But this method is not sensitive according to occurred order of events. Also, purely statistical outlier detection system may detect anomalies into normal data in less number of data with almost no variation. Therefore, the pure statistical method is used to a model that detects events are limited.

The following feature extraction is a way to extract a particular outlier detection pattern [11]. A set of heuristic tools and outlier detection will be set up. Intrusion detection and a subset of outlier detection tool are decided. Feature extraction can predict variously. When a new form of outliers is occurred, it may determine normal events into anomalies.

In the predictable pattern [13], an order of a specific event is not random and this method can be explained between order of events and correlation. Using Time-Based rules, it can assign an invent time into individual event. According to the time interval, it can detect whether given events is normal or not. For example, E1-E2-E3 pattern has occurred rate as 95% and E1-E2-E4 pattern is 5%. In this case, 'E3' is a normal event and 'E4' is an anomaly. Events like 'E4' that Occurred rate is low mean intrusion. Weakness of this method is not able to detect abnormal data if patterns are not defined in a rule. If a pattern is defined by an order, the pattern is handled to various ones. Abnormal events of a pattern are processed carefully.

Neural network detection [10] learns neural network and then it can predict unknown data based on a learned model. Strength of neural network is independent to data characteristics like statistical method and can detect well in data with much noise. It can also indicate correlation of various methods that influence output. Weakness of this method may be required so long learning time. When setting a size of current data and past data, size of data is difficult to be important factor on neural network design. If data size is massive, it can obtain exact result. However, it has long learning time. In contrast, if data size is less, it is difficult to obtain exact result. However, it has fast learning time.

3. ALGORITHM OF OUTLIER DETECTION

Our outlier detection model is consisted three steps. Section 3.1 is constructed basic models for outlier detection. Section 3.2 describes evolution process of micro-clusters. It consist grid maps. Section 3.3 is an application of grid maps.

3.1 Basic Constructs

Common outlier detection model was presented in 1987 by Dorothy Denning, the model is still used [9]. Most of the outlier detection system can build accurately an abstract model. And it is independent of types of input, observation system, and specific intrusion detection method.

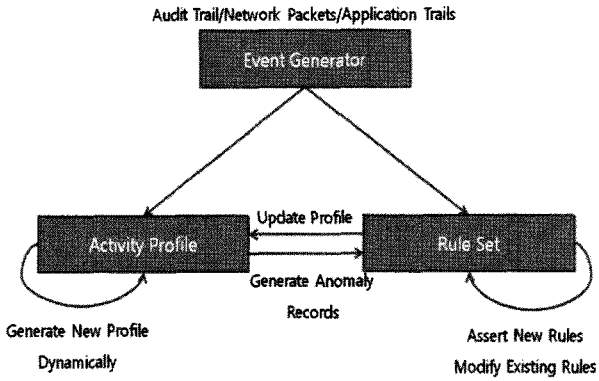

Figure 1. Outlier detection model

Figure 1 shows a typical outlier detection model. Event generator creates events in audit trail, network packet and application trail. Audit provided by a general audit daemon in the operating system records whole events occurred by a system.

Outliers of packets are detected by Network Packet on networks. Application Trail uses a tool in an existing operating system and generates events. Activity Profile represents the entire state of intrusion detection model and has variables required in a system using pre-defined statistical methods. Pattern Template generates new profiles of Subject and Object at regular time. Rule set is defined to the general inference mechanism. It updates their status and controls the actions of other components using event records, anomaly records, and the time.

Outlier detection model is required an advanced model for processing data streams. Processing continuous information is very important on data streams. Multidimensional data stream is incoming through the event generator. Rule set classify whole data into a predictable training data and a test data. When each data ($X_1 \ldots X_k \ldots$) is entered, the arrival time of each data is ($T_1 \ldots T_k \ldots$). D-demensional data set is $X_i = (x_{i1} \ldots x_{id})$. Proposed technique is adaptive classification used to fixed grid in terms of a data stream.

First, X-axis of two-dimension coordinates decides time axis and Y-axis is axis of value. X and Y-axis are divided into unit of fixed grid.

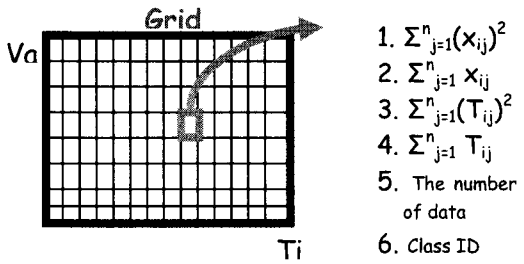

Figure 2. Characteristics of a micro-cluster

Figure 2 shows 6 characteristics of a micro-cluster. We can look for one micro-clusters which is divided into X and Y-axis. Micro-clusters are extended in "On Demand Classification". This micro-cluster has following characteristic that is summarized with "On Demand Classification". First, multidimensional attribute indicates the sum of X_i's square. The second attribute indicates X_i's sum. The third attribute indicates the sum of T_i's square. The fourth attribute indicates T_i's sum. The fifth attribute indicates the number of data on multidimensional environment. Last attribute indicates a class identification number.

Micro-clusters which store this information that is summarized are built CF-tree basically. CF-tree is easy to update the information and sort data that is summarized when CF-tree inserts and deletes new training data in a micro-cluster. Also, when a micro-cluster is merged with other micro-clusters, it can perform '+' operation by definition of CF-tree.

Micro-clusters have six characteristics and one Time Window. Time Window takes charge of increase and decrease of micro-clusters. Time Window that indicates by form of table consists of Frame's number, classifying with Snapshots (by clock time), and storing Time of Arrival of data. The insertion and deletion policy of Time Window are done equally with "On Demand Classification".

Classification of micro-clusters which use training data is excellent when adjacent data are inserted within fixed range. Defined training data is entered continuously individual time. The extent of clusters is changed by the extent change of training data.

3.2 Evolution Process

Micro-clusters building by training data classify experimental data. Experimental data can become communication network signal, bank data signal, network outlier signal or stock market change that we already have known. These data always do not consist of data within fixed range, as well as may be changed in dynamic range in real-time. This change can cope with model's evolution.

Model's evolution is variation of micro-clusters which appears on individual time. Micro-clusters changes in critical threshold that has defined by the user beforehand. If adjacent micro-clusters are existed within threshold, adjacent micro-clusters and current micro-cluster can be merging with same unique identification number.

In next step, adjusts similar clusters that compute Sim (X, Y) that indicates a similarity of clusters. The similarity can be made with characteristics of defined data beforehand. This cluster has same class ID.

After micro-clusters have same class ID, it merges adjacent micro-clusters and then the update operation of CF-tree is performed. CF-trees of adjacent micro-clusters are merged a CF-tree. For example, since two micro-clusters are adjacent, two micro-cluster X_is and X_j can merge. If it satisfy "Sim (X_i, X_j) >= Φ", X_i and X_j can be merged a micro-cluster which has same class ID.

To distinguish adjacent micro-clusters, it uses simple method. Standard range of threshold becomes unit of cell. That is, if threshold is a value as '1', it adjusts similarity of 8 micro-clusters which are closed to a micro-cluster.

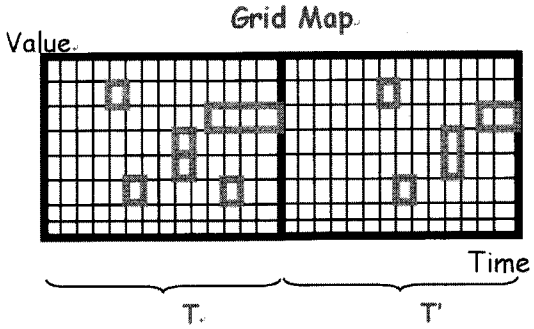

Figure 3. Evolution of adaptive micro-cluster

Figure 3 shows the evolution of micro-clusters by timing state. Micro-clusters which is clustered on current time T is merged in T' (T+ Δ t). The state of T' shows the evolution process of classification. Our model does not perform whole process of classification against all data and is only built in current and previous state.

It always performs creation, merger, and removal in order to build the model on data stream environments. The change of training data is related to evolution of clusters at individual time. Clustering technique based on fixed grid can analyze spatial area based on finite cells until arbitrary time variable Δ t at current time T. These cells are a space units that are performed all operations for clustering.

3.3 Process of Application

A data stream from various sources is gathered in the real world. Event Generator processes easily these data by category. These data can be processed in multidimensional outlier detection. As multidimensional outlier detection, it generates the grid maps and then training data are passed on a grid map of individual dimension according to importance of data.

Figure 4 is a process to detect outliers using multidimensional grid maps. When test data are gone through a grid map, the data are passed again in a next grid map if the data do not contain in a model. For

example, grid maps can be detected 3-dimensional input data as temperature, humidity and illumination. A grid map in top level is constructed micro-clusters about temperature. A grid map in second level is constructed micro-clusters about humidity. A grid map in bottom level is constructed micro-clusters about illumination.

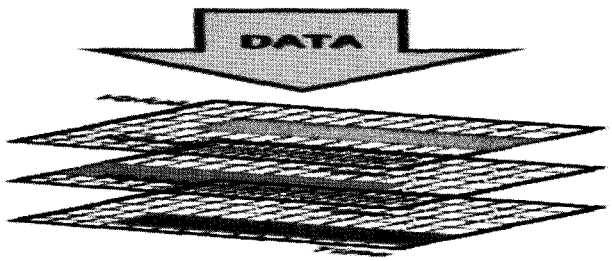

Figure 4. Process of Multidimensional grid maps

Outliers relevant to temperature are detected to a grid map in top level. And two remainders are through in a second grid and it detects outliers relevant to humidity. Outliers of illumination are detected in bottom level. Activity Profile updates information of CF-Tree and stores data extracted to CF-Tree. Data are through and detected anomalies in whole grid maps. Proposed grid map can detect critical outliers according to an order of priority.

4. CONCLUSIONS

We proposed an adaptive outlier detection method evolving based on fixed grid method. Statistical information stored in individual micro-cluster indicates summary information of data. This structure is efficient to perform parallel processing and incremental evolvement. Also, we ensure higher accuracy than fixed classification model according to temporal factor. In future work, we will implement the proposed technique and evaluate the performance of proposed method. It can be utilized in an application such as network outlier detection, stock market analysis, and error data detection in bank account.

References

[1] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, Jennifer Widom., June 2002, "Models and Issues in Data Stream Systems," Proc. of the 22th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Madison, Wisconsin, USA, pp. 1-16.

[2] C.C. Aggarwal, J. Han, J. Wang, and P. Yu, Aug 2004, "On Demand Classification of Data Streamsm," Proc. ACM KDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 503-508.

[3] C.C. Aggarwal, J. Han, J. Wang, and P. Yu, Sept. 2003, "CluStream: A Framework for Clustering Evolving Data Streams," Proc. Int'l Conf. Very Large Data Bases, pp. 81-92.

[4] Yunyue Zhu, Dennis Shasha., Aug 2002, "StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time," Proc. of the 28st Int'l Conf. on Very Large Data Bases, HongKong China, pp. 358-369.

[5] Liadan O'Callaghan, Adam Meyerson, Rajeev Motwani, Nina Mishra, Sudipto Guha., Feb 2002, "Streaming-Data Algorithms for High-Quality Clustering," Proc. of the 18th Int'l Conf. on Data Engineering, San Jose, California, USA, pp. 685-697.

[6] Charu C. Aggarwal., June 2003, "A Framework for Change Diagnosis of Data Streams," Proc. of the ACM SIGMOD Int'l Conf. San Diego, California, pp. 575-586.

[7] Tian Zhang, Raghu Ramakrishnan, Miron Livny, June, 1996, "BIRCH:An Efficient Data Clustering Method for Very Large Databases," Proc. of the ACM SIGMOD Int'l Conf., Montreal, Canada, pp. 103-114.

[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, March 1999, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Proc. of the second Int'l Conf. on Knowledge Discovery and Data Mining, Portland, Oregon, pp. 226-231.

[9] Dorothy E. Denning, Feb 1987,"An Intrusion-Detection Model", In IEEE Transactions on Software Engineering, Number 2.

[10] Kevin L. Fox, Ronda R, Henning Jonathan H. Reed, and Richard Simonian, Oct 1990, "A Neural Network Approach Towards Intrusion Detection", In Proceedings of the 13th National Computer Security Conference.

[11] A.H. Sung, and S. Mukkamala, 2003, "Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks", Proceedings of the 2003 Symposium on Applications and the Internet.

[12] Nong Ye, 2001, "A Scalable Clustering Technique for Intrusion Signature Recognition", Proceedings of the 2001 IEEE Workshop on Information Assurance and Security.

[13] T.D. Garvey and T.F Lunt, Oct 1991, "Model based Intrusion Detection", In Proceedings of the 14th National Computer Security Conference.

Acknowledgements

This work was supported by a grant from the Personalized Tumor Engineering Research Center (PTERC) and the Korea Science and Engineering Foundation(KOSEF)