International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

December 2014 | Issue 21 | Volume 6 | Number 4 | Pages 112-121

A PARALLEL ALGORITHM FOR ANOMALY DETECTION USING DATA CLUSTERING ON GRID

M. Khodizadeh Nahari

IT and Computer Engineering Department, Azarbaijan Shahid Madani University, Tabriz, Iran
m.khodizadeh@azaruniv.edu

Abstract- Anomaly in a set of data is an observation that is considerably dissimilar or inconsistent with the remainder of the data. Mining of anomalies has several applications in many areas and there exist different methods for anomaly detection, including data mining. Clustering is an unsupervised method for data mining that partitions data set into some clusters; anomalies are clusters with low density and long distance from others. In this paper, we propose a modification of distributed k-window algorithm that can achieve high-quality results in distributed computing environments such as Grid. To overcome the difficulty of anomaly detection in distributed data, we modify the base algorithm to present a new and efficient method. In modified algorithm, data transfer time and required storage are reduced and accuracy of data mining is increased.

Keywords: Anomaly Detection, Outlier Detection, Clustering, Grid Computing, Data Mining.

I. INTRODUCTION

While there is no single, generally accepted, formal definition of an outlier/anomaly, Hawkins' definition captures the spirit: "an outlier is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" [1].

An outlier/anomaly is defined as a data point that is very different from the rest of the data based on some measure. Such data often contains useful information on abnormal behavior in the system that is characterized by the data. The anomaly detection technique finds applications in credit card fraud, network intrusion detection, financial applications, and marketing. [2, 3, 4, 11, 12]. Many data-mining algorithms in the literature find outliers as a by-product of clustering algorithms [6, 7, 10, 20, 28]. However, these techniques define outliers as points that do not lie in any clusters. Thus, the techniques implicitly define outliers as the background noise in which the clusters are embedded.

Some literatures [5, 12, 14, 24, 25, 27] defines outliers as points that are neither a part of a cluster nor a part of the background noise; rather they are specifically points that behave very differently from the norm. Anomalies are more useful based on their value for determining behavior that deviates significantly from average behavior. In many applications such records may provide guidance in discovering important anomalies in the data. Such points are referred to as strong outliers in the work discussed in [25].

In fact, the identification of outliers can be applied in the areas of electronic commerce, credit card fraud detection, analysis of performance statistic of professional athletes [14] and even exploration of satellite or medical images [16].

Databases that are used in retailers, contains sale transactions, most transactions would involve a small amount of money and items. Thus typical outlier detection can discover exceptions base on some parameters such as: the amount of money spent, type of items purchased, time and geographical location of transaction .Nowadays, satellite image are uses to detect targets such as potential oil fields or suspicious military bases. Exceptions in detected energy or temperature or reflection of electromagnetic waves can be used to locate possible targets.

Outliers may have been symptom of anomalies. Stepaen Bay says: "Anomalies are occurrences of events that are unusual and that cannot be explained by our current knowledge of how the domain works. I see anomaly detection as simply trying to find these events. Anomalies are distinct from outliers, which I see as objects that on the measurement space are far away from the other points".

Dragos Margineantu says: "Anomalies are observations (or series of observations) with very low like hood of occurrence with respect to (1) the model(s) that are believed (or likely) to generate all observations and (2) the other observations that are available".

Decision support systems are used widely in decision making in many organizations, detection of outliers, avoids from wrong decisions, since abnormal data are discovered and discarded from data set that are used for decision making.

On the other hand, detected outliers might be the symptoms of a fraud or fault in a system. For example, in a normal case, traffic in a network has special characteristics, while pattern of traffic is nonstandard without any acceptable explanation, probably there would be an attack. In e-banking systems, each customer has a profile which is result of his/her behavior in a long period of time and describes normal behavior of him/her, if an outlier transaction is occurred, probably there would be a fraud.

Degree of variation is an important parameter that categorizes anomalies into three groups: (a) inconsistence data which has distance from the norm of data that must be considered, (b) outliers: that make problems and are more important than inconsistence data, (c) abnormal data which are fully incorrect and are dangerous.

Detecting algorithm works either: (a) by using an explicit model (such as mathematical model) that partitions normal and abnormal patterns, or (b) by continuous monitoring on system operation which helps detecting normal and abnormal behavior.

From the other viewpoint, anomaly detection algorithms are categorized in two groups: (a) real-time detector which always checks system and reports or responds for anomalies as soon as possible, (b) periodical detectors that check the system on special time.

Many techniques are applied for anomaly detection, e.g. (a) rule-based expert system, (b) statistical techniques, (c) data mining methods, (d) neural network, genetic algorithm and other AI techniques.

There are some parameters to detect anomalies such as: (a) time space between two events (b) normal event localities (c) resource usage (d) important counter. In a normal case these parameters have standard values which show regular state; if different values of these parameters are appeared they would be a mark of anomalies.

The rest of this paper is organized as follows: related work is summarized and discussed in section II. Clustering and its application in anomaly detection can be found in section III. The base algorithm and dataset are presented in section IV and our contribution can be found in section V. At the end of this paper conclusion and future works is presented.

II. RELATED WORKS

Most studies about outlier detection are in the field of statistics. A comprehensive study appears in Barnet and Lewis [5]. They provide a list of about 100 tests for detecting outliers in well-known unvaried distributions of data. However, Real-world data are commonly multivariate with unknown distribution.

Detecting outliers - instances in a database with unusual properties - is an important data mining application. People in the data mining community got interested in outliers after Knorr and Ng [17] proposed a non-parametric approach to outlier detection based on the distance of an instance to its nearest neighbors. Frequently, outliers are removed to improve accuracy of the estimators. However, this practice is not recommendable because sometimes outliers can have very useful information. The presence of outliers can indicate individuals or groups that have behavior very different from a standard situation.

According to various applications of anomaly detection, many different works are done in this area. Fraud is a dangerous type of anomaly. Detecting this type of anomalies is very important in commercial system. Lek et al. [9] have presented a research that use data mining techniques to generate rules from fraud patterns in e-commerce auditing systems. Subsequently, the system applies these rules to e-commerce databases with the aim of isolating those transactions that have a high chance of being fraudulent.

Chan et al. [10] have proposed some methods for fraud detection based on a "cost model". Their empirical results demonstrated that they can significantly increase benefit through distributed data mining of fraud models. The purpose of [11] is to study state of the credit card industry, different types of frauds and how a comprehensive fraud detection system could help maintain the cost of detecting fraud. Five of the most highly acclaimed data mining tools are so compared on a fraud detection application, with descriptions of their distinct strengths and weaknesses [14].

Steinwart and et al. in [15] detect anomalies by classification data into different layers. Aggarwal and Yu [16] introduce a new method for detecting anomalies in sparse dataset. The main problem in such dataset is the great distance between elements. Large dataset is the main consideration of many researches, Knorr and et al. in [17] present two distance-based algorithms, both having a complexity of $O(kN^2)$, where k is the dimensionality and N is the number of objects in the dataset. In the other research [18], Knorr and Ng work on large dataset with many different instances and attributes. A comprehensive survey is done in [19]. Its focus is on data mining based fraud detection techniques and real-life dataset accessibility problems.

High dimensional and sparse data is considered in [20, 21]. A workshop report about data mining methods for anomaly detection is prepared in [22]. Fraud detection by using association rule in web advertisement is discussed in [23].An iterative and unsupervised method in [24] attempts to detect anomaly. Reference [25] uses statistical techniques and evaluates his results on stock market data. In [26] Graph-based algorithms are applied on network traffic data. Early detection of illegal insurance claims discussed in [27].

III. CLUSTERING

Clustering is one of the fundamental methods for data mining which is the process of knowledge discovery from large databases. For example clustering means division of nodes to several groups in a sensor network [12]. Clustering is the partitioning a set of patterns into disjoint groups where members of a group are similar to each other and different from other group's member. These groups are called cluster. Figure 1 shows clusters which are detected in a sample dataset and some candidate for anomaly. Density of clusters and their distance from other clusters are two important parameters for anomaly detection so the small sized and distanced cluster are marked as an anomaly. Our work is based on this definition.

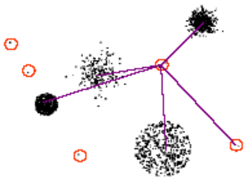

Figure 1. Clustering and anomaly detection

IV. ALGORITHM AND DATASET

There are many algorithms for anomaly detection. Most of them are applied on centralized datasets, but nowadays the size of data in databases is huge and databases have been distributed on several sites. K-Window is an algorithm which consists of three phases. The multi-phase nature of this algorithm leads us to apply it on a distributed environment (sub-section A, B). In this paper, some datasets are used for evaluating proposed algorithm that they are synthetic data. A simple program has been developed to generate data sets (sub-section C).

A. K-Window Algorithm

K-window is an algorithm that tries to place *n*-dimensional window (frame, box) containing all patterns that belong to a single cluster; for all clusters which are present in the dataset [6].

At first stage, some random windows are defined. These windows are moved in the Euclidean space without altering their size. Each window is moved by setting its center to the mean of the patterns currently included. This process continues iteratively until further movement does not increase the number of patterns included. At the second stage, the size of each window is enlarged in order to capture as many patterns of the cluster as possible. The process of enlargement ends when the number of patterns included in the window no longer increases. The two processes are exhibited in Figure 2, where the initial 2-dimensional window M1, is successively moved, and then subjected to two phases of enlargement that result to the final window E2.

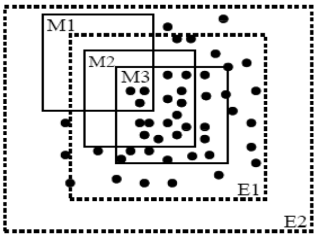

Figure. 2. Sequential Movements (solid lines) and subsequent, enlargements (dashed lines) of the initial window M1 that result to the final window E2 [6]

After the clustering procedure is terminated, the windows that are suspected to capture patterns that belong to a single cluster are merged. The merging procedure is illustrated in Figure 3. Windows W1 and W2 share a sufficiently large number of patterns between them, and thus, are merged to form a single cluster. On the other hand W5 and W6 are not merged, because although they overlap, the patterns in their intersection are insufficient to consider them as part of one cluster.

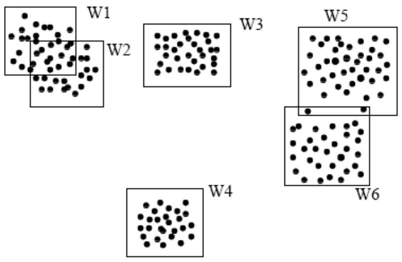

Figure 3. Merging procedure [6]

B. Parallel K-Window Algorithm

In a distributed computing environment the dataset is spread over a number of different sites. Thus, let us assume that the entire dataset X is distributed among m sites each one storing X_i for $i = 1, ..., m$, so .

$$X = \bigcup_{i=1,\dots,m} X_i \qquad (1)$$

Moreover, suppose that a central site (O) will hold the final results of clustering. The proposed implementation will assume that the data are distributed in m sites and the two initial phase of the algorithm is executed locally on the data. In other words, at each site i, the k-windows algorithm is executed over the X_i dataset. The result of these phases is a set of windows W_i for each site.

To obtain the final clustering result, all detected windows in each site are collected to the central node O. Central site performs merging and build the final clusters. In this phase all overlapped windows are considered for merging. The merge operation is based on the number of patterns that lie in the windows intersections.

In a distributed environment, the number of patterns in windows intersection may be unknown because of decentralized data or a site may not be willing to disclose this information. On the other hand, transfer large amount of data is a major obstacle. A high level description of the proposed algorithm is presented in Figure 4.

```
1. for each site $i$, with $i$=1,…,$m$
     execute the k-window algorithm over $X_i$
     send $W_i$ to the central node $O$
2. At the central node $O$
   for each site $i$
      get the resulting set of $d$-ranges $W_i$
      set $W \leftarrow W \cup W_i$
3. for each $d$-range $W_i$ not marked
      mark $w_j$ with label $w_j$
      if $\exists\ w_i \neq w_j$ that have sufficient overlap
      with $wj$ then
            mark $w_i$ with label $w_j$
```

Figure 4. Distributed k-window algorithm [6]

C. Dataset

In many cases, synthetic dataset is more suitable than authentic data for testing and training of anomaly detection systems. However synthetic data suffers from some drawbacks originating from the fact that they are synthetic and may not have the realism of authentic data. Using authentic data is not always a possible solution for evaluation of anomaly detection systems. For many applications, especially services that will be launched later, authentic data may not exist or may only be available in small quantities. In these situations, synthetic data is the only possible solution for conduction tests.

On the other hand, after applying anomaly detection algorithm, we must evaluate rate of success. In real dataset can't be determine anomalies easily but in synthetic dataset anomalies are inserted into dataset and after anomaly detecting, system can compare result with inserted anomalies.

Using synthetic data for evaluation and testing gives several advantages compared to using authentic data. Data properties of synthetic data can be tailored to meet various conditions; this property is not available in authentic datasets [8].

Synthetic data can be defined as data that are generated by simulated user in simulated system, performing simulated actions. There are some constraints and rules about synthetic data: The number of anomalies (ratio of abnormal/normal data) should be realistic and anomalies in input data are integrated with background data realistically. There is a great deal of complexity in synthetic data generation. Therefore a methodology is needed to structure the work and to point out the choices that have to be made [8].

V. CONTRIBUTION AND EXPERIMENTS

The main part of our work is related to modifications which we applied on the base k-window algorithm and distributed version of it.

A. Modifications Applied on Base Algorithms

Nowadays, the distribution of data is common. For anomaly detection in traditional approaches, we must collect the data in a central site and then apply the anomaly detection algorithm to the data. In our algorithm instead of collecting data in one site we use a distributed algorithm to detect anomalies on several sites.

To achieve this, we employed the distributed k-window algorithm (discussed in section IV), which consists of 3 phases: (a) clustering of data initially and movement, (b) cluster enlargement, (c) merging of clusters. In the base distributed k-window, phases 1, 2 are applied on the distributed data and on different sites then the resulted clusters are sent to central site for possible merging.

The base distributed algorithm has some problem when faces large and distributed datasets. To overcome the problems we modified the base algorithm. We called this algorithm: Rapid Distributed K-Window (RD K-Window).

A.1. Using Anomaly Definition

In anomaly detection process, clusters which have large sizes can't be marked as anomaly. Therefore instead of sending all detected clusters from phase 1 and 2, only small size clusters are sent to central node. In the central site, a merge algorithm only tries to combine small clusters. Clusters that are merged together and make larger ones are not candidate for anomaly. This method decreases network bandwidth requests and the data transfer time and data mining is done in high accuracy level, because huge amount of data isn't candidate for anomaly and is omitted from data sets. Finally the required storage space in the central site is decreased too.

A threshold is specified to highlight the small clusters. In the proposed model, initially we set the threshold value to a value which is presented by an expert. During iterations of algorithm running, system learns more about threshold value.

By using anomaly definition, processing of each data set is done in each site separately, and then anomaly candidate will be sent to central sites.

A.2. Arranging Central Nodes in a Hierarchical Structure

Hierarchical clustering is one the most used algorithm in many areas such as sensor network [13]. In base distributed k-window algorithm there are 2 levels processing nodes. First level nodes run movement and enlargement phase and second level nodes merges detected clusters in level 1.

Using Hierarchical structure of central nodes causes to detect anomalies in an evolutional process. This structure causes merging phase is done in several steps and in each step a few clusters are merged together.

In hierarchical structure, clusters which are merged with each other at level 2 can't be anomaly because they form a large enough cluster and won't be sent to higher level. This fact highlights the mentioned advantage, by distributing process among central sites.

This method helps us to use processing power of intermediate central sites and highlights the advantages are presented in the previous sub section.

A.3. Selecting the Initial Windows Intelligently

The base algorithm selects some random windows at the first stage. We use a suitable distribution mechanism to select the initial windows. In this mechanism (Figure 5), the selection isn't a pure random method; we divide data set into several sections and then select a initial window from each section so the clusters won't be encountered with some problems such as having a lot of common elements.

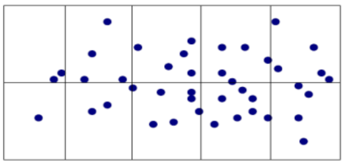

Figure. 5. Initial window selection

A.4. Detection of Data Clusters with Irregular Shapes

There is no accurate information about distribution of data so detection of clusters with irregular shapes is essential for these data. For this purpose in the enlarging phase, the windows are growing from each side up disjointedly. In some areas, these growths may be larger than other areas, so the final shape of the cluster probably will be irregular (Figure 6).

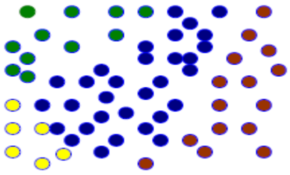

Figure 6. Cluster with irregular shape

A.5. Increasing Stability Encountering Noises

During enlargement phase, a new mechanism are used for alarming purpose, and the algorithm doesn't fail; when it faces to the first abnormal data, an alarm is sent and if the previous behavior is repeated the enlargement phase is stopped and algorithm looks for another cluster (Figure 7).

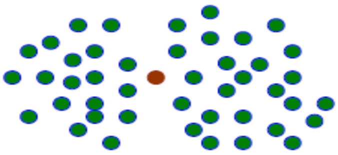

Figure 7. Weak noises

A.6. Prevention of High Overlapping of Clusters in Enlargement Phase

If in the enlargement phase a cluster entered into a domain which was occupied with another cluster, the growing won't continue, because the growth of this cluster wouldn't be more useful (Figure 8).

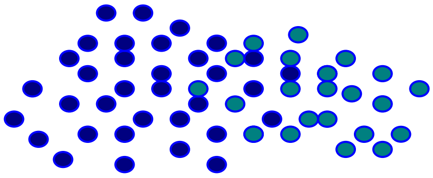

Figure 8. High overlapped window

A.7. Adding New Merging Phases to Base Algorithm

In the base algorithm, only clusters that are overlapped with others will be merged whereas in distributed data because of their non-centralized characters, related data aren't gathered in a cluster and overlaps are not observed.

According to section A.1 (using anomaly definition) only clusters with low density and long distance from the others, are sent to central sites so some overlapping cannot be detected.

So, a new kind of merging is a necessity for distributed data. This idea is done by sending the borders of omitted clusters to next level in hierarchical structure and calculating distance between two clusters base on these boundaries to specifying the amount of closeness or overlapping of clusters in the central sites. In this method cluster boundary should be determined after the formation of clusters in each level of hierarchical structure.

In this way, three kind of merging are needed: (a) Merging based on overlapping amount in the first phase and in the low level hieratical nodes. This type of overlapping could be detected easily because of all data are in a local site. (b) Merging based on distance of borders of clusters in the central nodes (Figure 9).

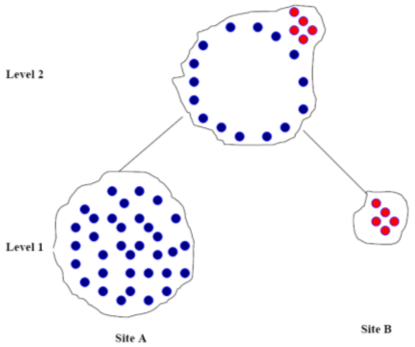

Figure 9. Merge base on border distance

(c) Merging based on overlapping of a cluster with the other one in the central nodes (Figure 10).

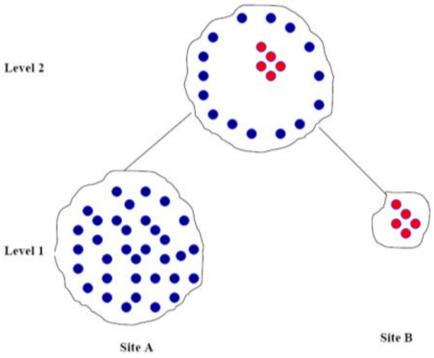

Figure 10. Merge base on overlapping

A.8. Calculating the Distance of Two Clusters Base on the Cluster Borders

In the existing method for calculating the distance of two clusters, the highest or closest distance of two clusters is considered as criteria. The average distance of all points of two clusters is more accurate than these two methods, but this method is very time consuming. So, usually a subset of points of two clusters is selected and the average distance of these two subsets are compared.

The selected subset in our study was the borders of the clusters. Principally the use of this distance factor between clusters for merging in the local sites is not very useful because, all data is centralized in a site and detection cluster overlapping is very easy as are done in the base algorithm. So this method suitable except for first level in hierarchical structure in distributed environment

A.9. Calculating the Overlapping of Clusters Base on the Cluster Borders

Clusters gathered from different sites may have overlap with each other, and because of data distribution, they cannot be combined through a simple merging method; following definition is used to overcome this problem:

In a n-dimension space, the point $A(x_1, x_2, \ldots, x_n)$ that is a member of cluster CA is considered as member of other cluster, CB, when there is a point B member of cluster CB that all its dimensions are greater than or equal to point A and there is a point C (other member of CB) that all its dimensions are smaller than or equal to point A. For example in a 2 dimensional space, if there are two points in cluster *X* that dimensions of one of them is smaller than or equal to point A, and the other with dimensions greater than or equal to point A, so the point A is inside cluster *X*.

A.10. Merging Anomaly Candidate Clusters with Large Clusters in Lower Layer

Because only clusters with low density and distanced from each other are sent to higher layer as anomaly ones so there is no chance to study the possibility of merging of this clusters with omitted big clusters, whereas anomaly candidate clusters may be merged with them and so never be accounted as anomaly. Therefore it is necessary to send some information of these large clusters to higher layer. So only the border point of large clusters are sent to higher layer (sub section A.7).

B. Running Modified Algorithm on Grid

The recent explosion of commercial and scientific interest in the Grid makes it timely to consider the question: What is the Grid, anyway? Carl Kesselman and Ian Foster attempted a definition (in 1998) in the book "The Grid: Blueprint for a New Computing Infrastructure.": "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." [28].

They refined the definition to address social and policy issues (in 2000), stating that Grid computing is concerned with "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations." The key concept is the ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose. They noted:

"The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization.". They also spoke to the importance of standard protocols as a means of enabling interoperability and common infrastructure.

Grid architecture has facilitated its extensibility. There are various service providers that provide required Grid services in a way that the users of these services easily use it in the solution that they have chosen, without aware of their details and complications. This is a great motivation to develop Grid base systems.

B.1. Design of Grid

To increase flexibility and processing power modified algorithm was implemented on a Grid that was established on a local area network with 350 node which was running Microsoft Windows 7. This Grid is a small size of a real Grid and can prove advantages of our algorithm. Our full ownership and control on this Grid is the main criteria for choosing it.

In this Grid, There is a node that has some useful information about the Grid. This node doesn't any control on Grid nodes and only delivered public information to others. This node is called Public Relation Node (PRN). Each node in local area network can connect to PRN and register to Grid. A software agent who is called "Grid Agent" is delivered to the Grid members when they are registered. All requests and services are delivered by using this agent.

Some nodes are registered as resource broker that we called them Resource Locator Node (RLN). These nodes may have a Relational Database Management System (RDBMS) such as MS-SQL Server to store some data and interpret SQL commands or a simple file management system. When a node has some sharable resources, it connects to a broker and introduces the resources (by using Grid Agent). These nodes are called Resource Provider Node (RPN). Information about the sharable resources is stored in a table on a database in RLN. The table stores such information: IP address of resource provider, Resource Type, Resource available level, Cost of resource, Cost of brokering.

When a node requires a resource that hasn't it, connect to a broker and request the resource (by using Grid Agent). We called these nodes Resource Requester Node (RRN). Some information about resource requests are stored in RLN such as IP address of requester node, Requested resource, Amount of requested node, Deliver time of requested resource, Provider node. These nodes is presented in Table 1.

Table 1. IP address of requester node

Agent Type	Abbr.
Resource Requester Node	RRN
Resource Provider Node	RPN
Resource Locator Node	RLN
Public Relation Node	PRN

B.2. Using the Grid to Anomaly Detection by Data Clustering

When the command of data clustering is issued, all agents receive a program that performs data clustering and the first phase of clustering algorithm (Parallel k-window) runs at each node. After extracting clusters, smaller clusters are sent to next level nodes in hierarchical structure as anomaly candidate.

Next level nodes are also a resource that are chosen by brokers and can be one of first nodes. The fact that which of the nodes are selected as high level nodes depends on their described power, for example these nodes must have execution code of cluster merging. The number of nodes that are connected to a high level nodes and the depth of hierarchy is dependent to Grid status and available nodes and free resource level of nodes. The scalability and availability of the system increase by using this method. Nodes doesn't have predefined structure in this method, however a very flexible and compatible with real needs structure is created due to situation.

A level number is assigned to each cluster at any level. When second level nodes received first package of data anomalies from the first level, they wait for a specific amount of time for other clusters. If in this period, specific amount of clusters are collected, the merging process will begin. If there aren't enough clusters for merging, the second level node will ignore the merging process and declares that it won't capable of doing this activity and informs its low level node about it. The low level node must looks for another resource provider. This rejection is allowed when there is another node in high level.

After doing the merging phase, the second level nodes do the same action to the third level nodes as first and second level nodes did together. The data clusters that get to the 3rd level have the second level label and totally separate from the first level data clusters. A node can have requests from the second and third level simultaneously; however this is when a sub tree of hierarchy has grown faster than the other sub trees.

This hierarchical structure can grow even more and stops when only one node is in last level. After executing data clustering, all nodes become aware of each other situation and the place of them and colleague nodes is determined (parent-child in the Rapid Distributed K-Window algorithm hierarchical structure)

B.3. Scheme Benefits

The nodes that have a lot of processing load are not fixed nodes so it isn't necessary to provide a lot of resources in them. Depend on conditions (e.g. traffic load) central nodes can be different nodes at each time. This can play a great role in improving the quality of services.

The scale of problem can be easily selected high, because in a good Grid there are always variant and great amount of resources that solve the problem. Because of great variety, the level of grid availability is high and the lack of a node doesn't make a lot of problem to the system. System throughput is high due to many resources are available and the response time is short.

As it seems, there is no central control and management over the network and there are only Brokers that play the role of intermediate between the providers and users of the resources and this causes the existing resources to be used optimally and huge operation and process are directed to more powerful nodes.

Resource providers can use variety of hardware and software. What is important is the physical connection with each other. Coordinated use of resource and utilizing standard protocols are the main important points that must be considered to increase the openness level of provided solution.

System does not impose any special default, thus the system does not need to be aware of the status of nodes or requests. The system doesn't have any control over nodes, thus they can be called an autonomous. The autonomy of nodes is too high so they can reject a request even if they can accept it.

According to Ian foster's checklist [28] a Grid is a system that:

(a) Coordinates resources that are not subject to centralized control: A Grid integrates and coordinates resources and users that live within different control domains

(b) Using standard, open, general-purpose protocols and interfaces: A Grid is built from multi-purpose protocols and interfaces that address such fundamental issues as authentication, authorization, resource discovery, and resource access.

(c) Delivers nontrivial qualities of service: A Grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts.

For example, the Web is not (yet) a Grid: its open, general-purpose protocols support access to distributed resources but not the coordinated use of those resources to deliver interesting qualities of service.

According to the above explanation, our provided system in this project is a Grid; because without applying centralized domination, it assigns resources to application in a coordinated form in a way that they don't become aware of the details exactly, in the other hand it uses standard protocols and the quality of services is high. The user of the Grid must think that he or she has a very powerful desktop, through this power has come from another place.

C. Dataset Generation

The performance of clustering algorithms partly depends on the characteristics of the data set. This section describes and discusses the dataset selected for the experiments.

For generation synthetic data we use profile mechanism. For example, in a financial application each customer has financial profile that describes his/her identification attributes and some financial information which presents financial characteristics of him/her.

In data generation phase a software module called "user/event simulator" generate normal data about one user/event. It uses user/event profiles that stored in a database. Other module called "abnormal data generator" mixes anomalies with normal data. The ratio of abnormal data/ normal data must be rational. In other words, distribution of normal and abnormal data must be realistic.

Synthetic data (mixture of normal and abnormal data) are stored in a database without any label on abnormal data. Also abnormal data are stored in other database separately. At the end of algorithm, detected anomalies are compared with this database.

There are three modules in this system: "user/event profiles", "User/event simulator" and "abnormal data generator" that models user/event. The process of data generation, anomaly detection and result evaluation are shown in Figure 11. The size of dataset was different in each time.

D. Experiments and Results

Our experiments are done in a local area network with 350 nodes. All of them are running on Windows 7 operating system. We use messaging system to coordinate distributed processes. We have developed all codes need to implement experiments.

A software module is developed to created synthetic dataset containing normal and abnormal data. This module prepares data for anomaly detector algorithm. In this paper, we generated 2-dimensional abstract data such as points in Euclidean space and integrated some abnormal data into that.

We use some metrics to evaluate performance of the algorithms. These metrics are Detection Time, Detection Rate, False Positive Rate. The results are shown for all experiments: traditional k-window algorithm, distributed k-window, modified algorithm in a dedicated hierarchical structure and on a Grid environment. All motioned times are in second.

In small dataset, the overhead time of using distributed version of algorithm is greater than execution time of algorithm; therefore in this case traditional algorithm is better than modified algorithm. The results in the other implementation (the modified distributed k-window and Grid base algorithm) are very excellent.

D.1. Time of Execution

The main metric for evaluation of performance is the execution time. Each algorithm consist of several phases that may be different from the others, for example, in traditional solution, time spends for finding cluster, cluster movement, merging cluster and anomaly detection whereas in our modified solution time spends for finding cluster, cluster movement, finding small size cluster, sending small size cluster to next level site, new merging phase and some other stages and finally anomaly detection.

For an end user, these details isn't important and total time between starting to finishing algorithm is critical. We also measured the time base on this fact.

D.2. Quality of Detection

Two important metric for our experiments are: false positive rate and false negative rate. Tables 2 up to 5 and Figure 12 show the run times, detection rate and false positive rate of base algorithm, base distributed algorithm, modified distributed and hierarchical algorithm and Grid based algorithm.

In Figure 13, we present the result of our contributions for this presentation. We remove our modification about algorithm speed-up and run our grid based algorithm. The results indicate the effect of modifications.

Table 2. Implementation of base algorithm

False Positive Rate	Detection Rate	Run Time(s)	Number of Records
3%	90%	76	3500
4%	88%	164	7510
6%	86%	280	12500
8%	88%	412	19750
9%	86%	886	35200
8%	85%	1249	85000

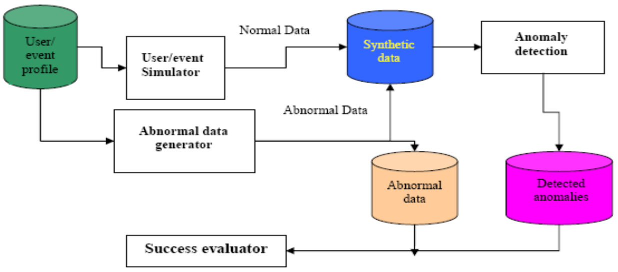

Figure 11. Process of data generation, anomaly detection and result evaluation

Table 3. Implementation of base distributed algorithm

False Positive Rate	Detection Rate	Run Time(s)	Number of Records
5%	89%	38	3500
3%	90%	45	7510
4%	87%	66	12500
6%	90%	112	19750
8%	86%	165	35200
7%	86%	326	85000

Table 4. Implementation of hierarchical and distributed

False Positive Rate	Detection Rate	Run Time(s)	Number of Records
2%	92%	18	3500
4%	88%	32	7510
5%	88%	48	12500
7%	90%	73	19750
6%	91%	102	35200
8%	87%	190	85000

Table 5. Implementation of algorithm on Grid

False Positive Rate	Detection Rate	Run Time(s)	Number of Records
3%	90%	25	3500
5%	89%	36	7510
5%	88%	45	12500
8%	89%	68	19750
7%	90%	99	35200
7%	88%	176	85000

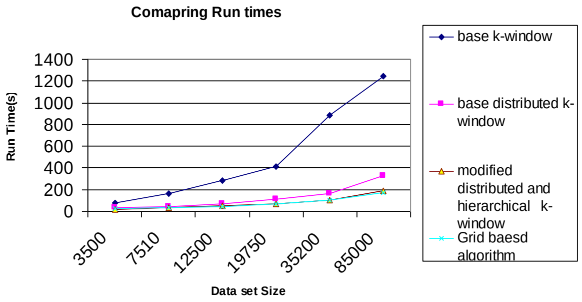

Figure 12. Comparing run times of 4 algorithms

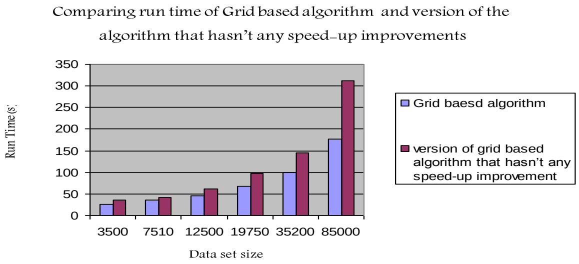

Figure 13. Comparing run times of Grid based algorithms

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have introduced a clustering algorithm and distributed version of it. Then we have modified it to detect anomalies in a large dataset. Instead of collecting data in central sites, we apply k-window algorithms on separate sites and send small size clusters in to central sites to merge together. We have made a hierarchal structure for central sites to merge together. The hierarchal structure reduces network bandwidth usage and time needed for anomaly detection. This method helps us to increase data mining accuracy and get better results.

Selecting the initial windows intelligently and detection of data clusters with irregular shapes has increased power of our algorithm. By using a new mechanism, stability of algorithm (when encounters with noises) is increased.

Other our works are: prevention of high overlapping of clusters in enlargement phase, adding new merging phases to the base algorithm, calculating the distance of two clusters base on the cluster borders, calculating the overlapping of clusters base on the cluster borders, merging anomaly candidate clusters with large clusters in lower layer and calculating anomaly coefficient.

Our experiments shows that instead of collecting distributed data in central site and applying clustering algorithm, distributed clustering achieve better result. Grid is the computing and data management infrastructure that will provide the electronic foundation for global society in business, government, research, science and entertainment [29]. Grid infrastructure is suitable to implement hierarchical structure dynamically to support the execution of large scale, resource-intensive, and distributed application.

Anomalies may be a symptom of a fraud, fraud detection especially in commercial domain is a new area to continue this study. In our modified algorithm, data transfer time and required storage are reduced and accuracy of data mining is increased.

Finding best value for threshold which discussed in section V and considering inconsistency in location and time of events is other works in this domain.

REFERENCES

[1] D. Hawkins, "Identification of Outlier", Chapman and Hall, London, 1980.

[2] V. Chandola, A. Banerjee, V. Kumar, "Anomaly Detection: A Survey", ACM Computing Surveys (CSUR)", Vol. 41, No. 3, 2009.

[3] E.W.T. Ngai, et al., "The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review of Literature", Decision Support Systems, Vol. 50, No. 3, pp. 559-569, 2011.

[4] P. Gogoi, et al., "A Survey of Outlier Detection Methods in Network Anomaly Identification", The Computer Journal, 2011.

[5] R. Pamula, K.K. Deka, S. Nandi, "An Outlier Detection Method Based on Clustering", Second IEEE International Conference on Emerging Applications of Information Technology (EAIT), 2011.

[6] D.K. Tasoulis, M.N. Vrahatis, "Unsupervised Distributed Clustering", IASTED International Conference on Parallel and Distributed Computing and Networks, pp. 347-351, 2004.

[7] M.N. Vrahatis, B. Boutsinas, P. Alevizos, G. Pavlides, "The New K-Windows Algorithm for Improving the K-Means Clustering Algorithm", Journal of Complexity, Vol. 18, pp. 375-391, 2002.

[8] E.L. Barse, H. Kvarnstrom, E. Jonsson, "Synthesizing Test Data for Fraud Detection Systems", 19th Annual Computer Security Applications Conference, pp. 384-394 2003.

[9] M. Lek, B. Anandarajah, N. Cerpa, R. Jamieson, "Data Mining Prototype for Detecting E-Commerce Fraud", Proceedings of the ECIS'2001, Bled Slovenia, 27-29 June, 2001.

[10] P.K. Chan, W. Fan, A.L. Prodromidis, S.J. Stolfo, "Distributed Data Mining in Credit Card Fraud Detection", IEEE Intelligent Systems, Vol. 14, pp. 67-74, 1999.

[11] T.P. Bhatla, V. Prabhu, A. Dua, "Understanding Credit Card Frauds", Tata Consultancy Services, June 2005.

[12] H. Sedghani, M. Zolfy Lighvan, "PDH Clustering in Wireless Sensor Networks", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 20, Vol. 6, No. 3, pp. 121-125, 2014.

[13] H. Barati, A. Movaghar, A.M. Rahmani, A. Sarmast, "A Distributed Energy Aware Clustering Approach for Large Scale Wireless Network", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 13, Vol. 4, No. 1, pp. 125-132, 2012.

[14] D.W. Abbott, I.P. Matkovsky, J.F. Elder, "An Evaluation of High-End Data Mining Tools for Fraud Detection", IEEE International Conference on Systems, Man, and Cybernetics, Vol. 3, pp. 2836-2841, 1998.

[15] I. Steinwart, D. Hush, C. Scovel, "A Classification Framework for Anomaly Detection", Journal of Machine Learning Research, Vol. 6, pp. 211-232, 2005,

[16] C.C. Aggarwal, P.S. Yu, "An Effective and Efficient Algorithm for High Dimensional Outlier Detection", International Journal on Very Large Database (VLDB), pp. 211-221, Springer Press, 2005.

[17] EM. Knorr, RT. Ng, V. Tucakov, "Distance Based Outliers: Algorithms and Applications", International Journal on Very Large Database (VLDB), Vol. 8, pp. 237-253, Springer Press, 2000.

[18] E.M. Knorr, R.T. Ng, "A Unified Approach for Mining Outliers", Conference of the Centre for Advanced Studies on Collaborative Research, p. 11, 1997.

[19] C. Phua, V. Lee, K. Smith, R. Gayler, "A Comprehensive Survey of Data Mining Based Fraud Detection Research", Artificial Intelligence Review, 2005, bsys.monash.edu.au.

[20] C.C. Aggarwal, P.S. Yu, "Outlier Detection for High Dimensional Data", ACM SIGMOD International Conference, pp. 37-46, 2001.

[21] C. Phua, D. Alahakoon, V. Lee, "Minority Report in Fraud Detection: Classification of Skewed Data", ACM SIGKDD Explorations Newsletter, Vol. 6, pp. 50-59, 2004.

[22] D. Margineantu, S. Bay, P. Chan, T. Lane, "Data Mining Methods for Anomaly Detection KDD-2005 Workshop Report", ACM SIGKDD Explorations Newsletter, Vol. 7, pp. 132-136, 2005.

[23] A. Metwally, D. Agrawal, A.El. Abbadi, "Using Association Rules for Fraud Detection in Web Advertising Networks", 31st International Conference on Very Large Databases VLDB, pp. 169-180, 2005.

[24] K. Yamanishi, J. Takeuchi, "Discovering Outlier Filtering Rules from Unlabeled Data", Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 389-394, 2001.

[25] S. Ramaswamy, R. Rastogi, K. Shim, "Efficient Algorithms for Mining Outliers from Large Datasets", ACM SIGMOD International Conference on Management of Data, Vol. 29, pp. 427-438, 2000.

[26] C.C. Noble, D.J. Cook, "Graph Based Anomaly Detection", Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 631-636, 2003.

[27] T. Ormerod, N. Morley, L. Ball, Ch. Langley, C. Spenser, "Using Ethnography to Design a Mass Detection Tool (MDT) for the Early Discovery of Insurance Fraud", Human Factors in Computing Systems Conference, pp. 650-651, 2003.

[28] I. Foster, C. Kesselman, "The Grid: Blueprint for New Computing Infrastructure", 2nd Edition, Morgan Kaufmann and Elsevier, Chapters 4 and 17, 2004.

[29] F. Berman, G. Fox, T. Hey, "Grid Computing: Making the Global Infrastructure a Reality", The Grid: Past, Present, Future, Chapter 1, pp. 9-50, John Wiley & Sons, 2003.

BIOGRAPHY

Mohammad Khodizadeh Nahari received the B.Sc. degree from Electrical and Computer Engineering Department, Isfahan University of Technology, Isfahan, Iran, in 1998 and the M.Sc. degree from Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran, in 2008. He joined Azarbaijan Shahid Madani University, Tabriz, Iran as Lecturer since 2009. He is a Ph.D. student in the Electrical and Computer Engineering Department, Isfahan University of Technology since 2012. His research are mostly concentrated on the data mining, data management system, data warehouse and bioinformatics.