# Anomaly Intrusion Detection based on Clustering a Data Stream

Jinsuk Kang and Sanghyun Oh

***Abstract*—This paper proposes a new clustering algorithm which continuously models a data stream. A set of features is used to represent the characteristics of an activity. For each feature, the clusters of feature values corresponding to activities observed so far in an audit data stream are identified by the proposed clustering algorithm for data streams. As a result, without maintaining any historical activity of a user physically, new activities of the user can be continuously reflected to the on-going result of clustering.**

***Index Terms*—Intrusion detection, anomaly detection, data mining, clustering, data stream.**

## I. Introduction

Recently, many data mining methods [1] for a data stream have been actively introduced. A data stream is an ordered sequence of objects $o_1, \ldots, o_n$ that must be accessed in order and that can be read only once or a small specified number of times. As a result, it is impossible to maintain all objects of a data stream in the main memory. Consequently, each object should be examined at most once to analyze a data stream. In addition, memory usage for data stream analysis should be confined finitely, although new objects are continuously generated in a data stream. So that it can be instantly utilized upon request, newly generated objects should be processed as quickly as possible in order to maintain an up-to-date analysis result of the data stream, so that it can be instantly utilized upon request. To satisfy these requirements, data stream processing sacrifices the correctness of its analysis results by allowing some errors. Meanwhile, clustering methods [2, 3] are suitable for modeling a large number of data objects as long as there exists a distance measure among them. This is because it is based on data similarity. Clustering is a process of partitioning a plain collection of data objects into meaningful groups called clusters. In other words, the purpose of clustering is to locate the groups of similar data objects which are defined by a given similarity measure. Consequently, the potential groups and structures of data objects in a data set can be identified.

In this paper, we propose an anomaly detection method based on clustering a data stream. In most conventional clustering methods on data streams, only a given number of clusters are identified. However, since the number of clusters in a data stream is unknown, their quality can be poor. On the other hand, in the proposed method, a cluster can be split into two clusters or two clusters can be merged into one cluster with respect to the distribution of objects occurring in a data stream. Therefore, clusters can be more effectively identified. In the proposed method, the various statistics of the objects in terms of identified clusters are modeled as a profile in order to improve the performance of anomaly detection. Whenever a new activity is performed, the profile is updated instantly. In addition, to evaluate the proposed method, synthetic data sets and 1998 DARPA data sets are used.

## II. Clustering Streaming Data

### A. Cluster Streaming

When multiple dimensions are associated with each object in a data stream, multidimensional clustering can be used to find a set of multidimensional clusters in a data stream. However, the number of dimensions i.e. features can be large in anomaly detection and each object in a data stream is not necessarily related to all the dimensions of the data stream. In other words, each object maintains a set of values for only related dimensions. Therefore, in the proposed method, anomaly detection is performed by using a one-dimensional clustering method. In most conventional clustering methods, the number of clusters is given in advance. However, since the number of clusters in a data stream is unknown, inaccurate clusters may be identified. Unlike conventional methods, in the proposed method, a set of clusters are dynamically identified with respect to the distribution of objects occurring in a data stream. For the purpose, each object is considered as a cluster until the number of objects occurring in a data stream is the same as a given *initial_cluster_number*. In the proposed method, a center point represents each cluster. Therefore, whenever a new object occurs, the object is inserted into a cluster whose center is closest to the object. As time goes by, the range of each cluster can become too large and several clusters can be very close to one another. Therefore, in order to maintain the quality of the clusters identified from a data stream, any cluster whose range is too large should be split into more than two clusters and very close clusters should be merged into one cluster.

A data stream is represented by $S = \{o_1, o_2, \ldots, o_h\}$ and an object $o_i$ is represented by n-dimensional vector i.e., $o_i = (o^1{}_i, o^2{}_i, \ldots, o^n{}_i)$. Therefore, a stream data projected to the $k^{th}$ dimension from the data stream S is represented by $S^k = \{o^k{}_1, o^k{}_2, \ldots, o^k{}_h\}$. Let $X^k$ denote a set of clusters identified from $S^k$. In order to effectively maintain the quality of the clusters, each cluster has a set of grid-cells. The interval of a grid-cell is defined by a non-overlapped and equal-sized unit. In order

Manuscript received March 14, 2012; revised May 6, 2012.

Jinsuk Kang is with the Jangwee Research Institute of National Defene, Ajou University, South Korea (e-mail: jskang01@ajou.ac.kr)

Sanghyun Oh is with the Java Inc. Corp, South Korea.

to represent a unit, only one value, *a unit identifier*, is used in this paper. For instance, an object $o^k$ is transformed to a unit identifier $I^k$ ($I^k = \lceil o^k/\rho^k \rceil$) where $\rho^k$ represents an interval size for the k<sup>th</sup> dimension. A grid-cell $g^k$ contains the following information: the unit identifier $I^k$, linear sum $gl^k$, square sum $gs^k$ and total number $gn^k$ of objects contained in the interval of $g^k$ i.e., $g^k = (I^k, gl^k, gs^k, gn^k)$. The properties of a cluster $C^k \in X^k$ are represented by Section 2.2.

### B. Cluster Properties

Given a cluster $C^k$ of similar data objects for the k<sup>th</sup> dimension, the properties of a cluster $C^k$ are represented by a tuple $C^k(\delta^k, \mu^k, SS^k, gSet^k)$. $\delta^k$: The density of the cluster $C^k$ is represented by $\delta^k$. It is set to the total number of objects in the cluster $C^k$. $\mu^k$: The central value of the cluster $C^k$ is represented by $\mu^k$. It is calculated by the average of objects in the cluster i.e.,

$$\mu^k = \frac{1}{\delta^k} \cdot \sum_{j=1}^{r} o_j^k \qquad (1)$$

$SS^k$: The square sum of objects contained in $C^k$ is represented by $SS^k$. It is calculated by the square sum of objects in $C^k$ i.e., $SS^k = \sum_{j=1}^{r} (o_j^k)^2$. $\sigma^k$: The standard deviation of objects in $C^k$ is represented by $\sigma^k$. It is calculated by $\sigma^k = \sqrt{SS^k/\delta^k - (\mu^k)^2}$. $gSet^k$: The grid-cell set of objects contained in $C^k$ is represented by $gSet^k$. When a new object $o^k$ occurs from a data stream $S^k$, a cluster whose center is closest to the object should be selected from $X^k$ in order to insert the object $o^k$ into the cluster. The updated properties of the cluster $C^k$ are calculated as follows:

$$C^k\left(\delta^k + 1, \frac{\mu^k \cdot \delta^k + o^k}{\delta^k + 1}, SS^k + (o^k)^2, gSet^k\right) \qquad (2)$$

To update the grid-cell set $gSet^k$ consider the only grid-cell whose interval contains the object $o^k$. In other words, when the unit identifier of a grid-cell $g^k$ is same as the unit identifier of the object $o^k$, the properties of the grid-cell $g^k$ can be updated. Let $\overline{gl}^k$, $\overline{gs}^k$ and $\overline{gn}^k$ denote the old properties of $g^k$ and let $gl^k$, $gs^k$ and $gn^k$ denote the new properties of $g^k$, respectively. Ultimately, for a grid-cell $g^k \in gSet^k$ whose unit identifier $I^k$ equals to $\lceil o^k/\rho^k \rceil$, the new properties of $g^k$ are calculated as follows:

$$g^k = \left(gl^k + o^k, gs^k + (o^k)^2, gn^k + 1\right) \qquad (3)$$

When many objects occur between two adjacent clusters, they are very close each other. If the standard deviation of all the objects in them becomes smaller than or equal to a user-defined threshold *minimum deviation*, they are merged into one cluster. For two adjacent clusters $C^k_1$ and $C^k_2$ contained in $X^k$, let $\sigma$ denote the standard deviation of all objects in the two clusters and it is calculated as follows:

$$\sigma = \sqrt{\frac{SS_1^k + SS_2^k}{\delta_1^k + \delta_2^k} - \left(\frac{\mu_1^k \cdot \delta_1^k + \mu_2^k \cdot \delta_2^k}{\delta_1^k + \delta_2^k}\right)^2} \qquad (4)$$

If $\sigma \leq$ *minimum_deviation*, the two clusters are merged into one cluster $C^k$. The properties of the cluster $C^k$ is updated as follows:

$$C^k\left(\delta_1^k + \delta_2^k, \frac{\mu_1^k \cdot \delta_1^k + \mu_2^k \cdot \delta_2^k}{\delta_1^k + \delta_2^k}, SS_1^k + SS_2^k, gSet_1^k \cup gSet_2^k\right) \qquad (5)$$

In the above equations, the center of the cluster $C^k$ is set to the weighted average of the centers of the two clusters $C^k_1$ and $C^k_2$ with respect to their densities $\delta^k_1$ and $\delta^k_2$. Also, the square sum of the cluster $C^k$ is set to the sum of $SS^k_1$ and $SS^k_2$. When the cluster $C^k_1$ and $C^k_2$ are merged to the cluster $C^k$, the grid-cell set $gSet^k$ of $C^k$ can be obtained by the union of their grid-cell sets $gSet^k_1$ and $gSet^k_2$, i.e., $gSet^k = gSet^k_1 \cup gSet^k_2$. This is because the grid-cells of two clusters are not overlapped according to the definition of a grid-cell.

Meanwhile, as the number of objects occurring from a data stream becomes larger, the standard deviation of objects in each cluster becomes higher. In other words, the quality of each cluster can be low. Therefore, to maintain the quality of each cluster, the cluster is split into two clusters with respect to *minimum_deviation* as follows. For a cluster $C^k \in X^k$, let $gSet^k$ be $\{g^k_1, g^k_2, ..., g^k_p, ..., g^k_q\}$. If $\sigma^k >$ *minimum_deviation*, then the cluster $C^k$ is split into two clusters $C^k_1$ and $C^k_2$. Let $T^k_i$ denote a set of objects contained in $g^k_i \in gSet^k$ and let $r^k_p$ denote the total number of objects contained in $\bigcup_{i=1}^{p} T_i^k$ i.e., $r_p^k = \sum_{i=1}^{p} gn_i^k$. If $r_{p-1}^k < \delta^k/2 \leq r_p^k < \delta^k$, then the grid-cell sets of the two clusters are $gSet^k_1 = \{g^k_1, g^k_2, ..., g^k_p\}$ and $gSet^k_1 = \{g^k_{p+1}, ..., g^k_q\}$. As a result, the properties of $C^k_1$ and $C^k_2$ can be obtained as follows:

$$\begin{gathered} C_1^k\left(\sum_{i=1}^{p} gn_i^k, \frac{1}{r_p^k}\sum_{i=1}^{p} gl_i^p, \sum_{i=1}^{p} gs_i^p, \bigcup_{i=1}^{p}\{g_i^k\}\right), \\ C_2^k\left(\sum_{i=p+1}^{q} gn_i^k, \frac{1}{\delta^k - r_p^k}\sum_{i=p+1}^{q} gl_i^k, \sum_{i=p+1}^{q} gs_i^k, \bigcup_{i=p+1}^{q}\{g_i^k\}\right) \end{gathered} \qquad (6)$$

Algorithm 1 describes the process of clustering a data stream. In this algorithm, input parameters are a data stream $S^k$, a minimum deviation, and an initial cluster number. In Line 1, each object is generated as a cluster with respect to the initial cluster number. And then, when a new object occurs, the following processes are performed. In Lines 3~4, a cluster whose center is closest to a new object is selected from the cluster set and the properties of the cluster are updated. In Lines 5~10, when the standard deviation of all the objects in any two adjacent clusters is less than or equal to the minimum deviation, two clusters are merged. In Lines 11~15, when the standard deviation of any cluster becomes larger than the minimum deviation, the cluster is split into two clusters and then newly generated clusters are inserted into the cluster set $X^k$.

**Algorithm 1. Clustering a data stream**

```
Clustering (S^k, minimum_deviation, initial_cluster_number)
1) Generate initial clusters w.r.t initial_cluster_number
   and then insert them into X^k.
2) foreach o ∈ S^k do
3) Select the closest cluster C^k in X^k from o.
4) Update the properties of the cluster C^k.
5) Select the most adjacent cluster C^k' of the cluster C^k.
6) Calculate the standard deviation σ of objects in two
```

```
    clusters $C^k$ and $C^{k}{}'$
7)  if $\sigma \le$ minimum_deviation, then
8)      Merge $C^k$ and $C^{k}{}'$ into $C^k$.
9)      Update the properties of the cluster $C^k$.
10)     Delete $C^{k}{}'$ from $X^k$.
11) if $\sigma^k >$ minimum_deviation, then
12)     Split $C^k$ into two clusters.
13)     Update the properties of the two clusters.
14)     Insert the two clusters into $X^k$.
15)     Delete $C^k$ from $X^k$.
```

## III. Anomaly Detection

For each feature, the on-going result of clustering is summarized in a profile, which is composed of the two properties of each cluster, a center and a standard deviation. An anomaly in a newly occurring object can be identified by comparing the new object with the current profile of each feature. For this purpose, as a new object occurs from a data stream, if the difference between the object and its closest cluster becomes large, this object is considered as an anomaly. The difference *diff($X^k$, $o^k$)* between an objects and its closest cluster $C^k$ is defined as follows:

$$diff(X^k, o^k) = \frac{|\mu^k - o^k|}{\sigma^k}(C^k \in X^k) \qquad (7)$$

In the above equation, the distance between the cluster $C^k$ and the object $o^k$ should be divided by the standard deviation $\sigma^k$. This is because the common characteristics of the features should be normalized. As a result, when the number of features participating in anomaly detection is *n*, the overall abnormality of a new object can be calculated as follows:

$$abnormality(o) = \frac{1}{n} \cdot \sum_{i=1}^{n} diff(X^i, o^k) \qquad (8)$$

In order to decide the rate of abnormal behavior in the new object o, a set of different abnormality levels can be defined relatively to the normal behavior of the historical activities. In this paper, two different abnormality levels (*green, red*) are considered in order to classify whether the activities of a new object are anomalous or not. The green level is safe while the red is warning. Let $\Upsilon(\nu, \lambda, \chi)$ denote the statistics of abnormalities until now. $\nu$, $\lambda$ and $\chi$ are represented as the total number of objects occurring from a data stream S, the linear sum of their abnormalities and the square sum of their abnormalities, respectively. Based on the statistics $\Upsilon$, the average $\Phi$ and its standard deviation $\Theta$ of abnormalities can be calculated as follows.

$$\Phi = \lambda/\nu, \Theta = \sqrt{\chi/\nu - \Phi^2} \qquad (9)$$

The new object *o* is in

- *Green level: if* $0 \le abnormality(o) \le \Phi + \Theta \cdot \xi$.
- *Red level: if* $\Phi + \Theta \cdot \xi < abnormality(o)$.

A detecting factor $\xi$ is a user-defined parameter which determines how strictly the anomaly of a new object is classified. As it is decreased, a new object is more strictly examined. Given a set of normal object, its false alarm rate is represented by the ratio of the number of objects that are within the range of the red level over the total number of normal objects. Similarly, given a set of anomalous objects, its anomaly detection rate is represented by the ratio of the number of objects that are within the range of the red level over the total number of anomalous objects.

## IV. Experimental Results

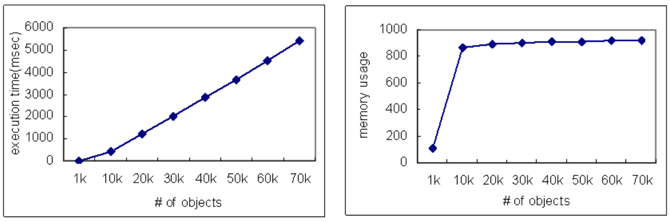
We present the results of experiments comparing the performance of LSEARCH and the proposed method. We conducted all experiments on a Pentium II with dual 350 MHz processors running LINUX 2.6.7. To demonstrate the performance of the proposed algorithm, synthetic data sets are generated as in table 1. The data sets D20, D40 and D80 contain clusters which are explicitly separated and the number of clusters in each data set is 20, 40 and 80, respectively. The data set RAN contains randomly generated objects. In all experiments in this paper, the interval of grid-cell and the minimum deviation are set to 5 and 30, respectively.

TABLE I: Synthetic Data Sets

| Data sets | # of objects | # of clusters | Data size |
|---|---|---|---|
| D20 | 11,600 | 20 | 51 Kbytes |
| D40 | 23,200 | 40 | 109 Kbytes |
| D80 | 46,400 | 80 | 225 Kbytes |
| RAN | 79,200 | unknown | 374 Kbytes |

TABLE II: Average SSQ for Each Data Set

| | D20 | D40 | D80 | RAN |
|---|---|---|---|---|
| Proposed Method | 70.4 | 72.9 | 78.1 | 550.8 |
| LSEARCH | 70.0 | 13399.2 | 44858.6 | 13834.3 |



(a) Execution time

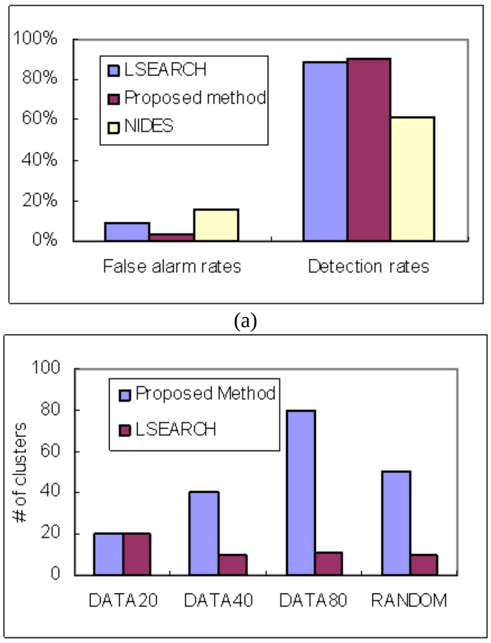
(b) Memory usage

Fig. 1. Performance of the proposed method

Fig. 1-(a) illustrates the execution time and memory usage of the proposed method when the number of objects in the RAN data set is varied. In Fig. 1, as the number of objects becomes larger, the execution time increases linearly. This is because the complexity of the proposed algorithm is O(n). Fig. 1-(b) illustrates the memory usage of the proposed method with respect to the number of objects occurring in the RAN data set. The memory usage is represented by the total number of grid-cells in clusters. In this experiment, the memory usage is saturated with about 900 when the number of objects is 1000.

Table I illustrates the average SSQ's of the proposed method and LSEARCH which is popularly used for representing clustering quality. Average SSQ is the average of squares of the distances to the cluster centers. In LSEARCH, the lower and upper bounds on the numbers of clusters to find are set to 10 and 80. For the data set D20, the average SSQ's of two methods are similar while those of LSEARCH are very higher than those of the proposed method for other data sets D40, D80 and RAN. It means that the proposed method finds clusters more correctly than LSEARCH. The reason of that result is shown in Figure 2(a). Figure 2(a) describes the number of clusters generated by the proposed method and LSEARCH. The proposed method finds correctly explicit clusters in data sets while LSEARCH do not.

In order to evaluate the performance of the proposed algorithm in a real world environment, we use DARPA log data sets collected in 1998 [4]. The feature values of the log data sets are extracted by BSM (Basic Security Module) [5] of Solaris 2.6. Among these signals, 84 signals are used as basic features in the experiments. In a log data set, an object is defined by the number of system calls occurring in a unix command on a host computer. We use two types of data sets for real world experiment: a programmer and a system administrator. A programmer writes a public domain C code via a “vi editor”, compiles the C code (sometimes successfully), reads and sends mails, and executes unix commands. A system administrator runs privileged commands. In this experiment, the programmer is regarded as a target user for anomaly detection. To simulate the environment of each data stream, a data set is replicated multiple times and its transactions are looked up one by one in sequence.



(a)

(b)

Fig. 2. The number of cluster (a) and Detection results

In Figure 2(b), the false alarm and detection rates in the proposed method are compared with those of the LSEARCH and NIDES. In this experiment, the value of the detecting factor ξ is set to 1.5. As shown in Figure 2(b), the false alarm rates of LSEARCH and NIDES are higher than that of the proposed method. Furthermore, the detection rate of NIDES is much lower than those of the proposed method and LSEARCH. As a result, the proposed method can detect an anomaly more effectively than LSEARCH and NIDES.

## V. Conclusions and Future Work

This paper proposes an anomaly detection method that employs a clustering algorithm for a data stream. For each feature, its clusters can be effectively found upon without maintaining any object of the data stream physically. For the purpose, clusters are dynamically generated by splitting a cluster into two clusters or merging two adjacent clusters into one cluster. As a result, the proposed method can find clusters more correctly than other conventional methods. For anomaly detection, new objects are continuously reflected to both the on-going result of clustering and the profile at the same time. Therefore, an anomaly can be detected easily without additional processes. In the future, the method of dynamically finding the interval size of a grid-cell and a minimum deviation is researched.

## Acknowledgements

This research was supported by the Basic Science Research Program through the NRF funded by MEST (No.2012-0005127) and (No.2011-0026536).

## References

[1] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, *Clustering data streams: Theory and practice. IEEE Trans. Knowl. Data Eng.* 2003, vol. 15, no. 3, pp. 515--528.

[2] J. Mac Queen, “Some Methods for Classification and Analysis of Multivariate Observations,” *Proc. 5th Berkeley Symp*, 1967, pp. 281-297.

[3] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, “Birch: An Efficient data clustering method for very large databases,” *Proceedings for the ACM SIGMOD Conference on Management of Data*, Montreal, Canada, 1996, pp. 1-10.

[4] http://www.ll.mit.edu/IST/ideval/index.html

[5] Sun Microsystems. SunShield Basic Security Module Guid.