



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

(Open Access, Double Blind Peer-reviewed, Refereed and Indexed Journal)

www.iasir.net

Meddling Detection in Pre Cluster Count in Unlabelled Dataset Using Virtual Assessment Tendency

S.Lingeswari, P.Vijayalaksmi

Assistant Professor, Department of Computer Science,
P.K.R. Arts College for Women, Gobichettipalayam,
Tamil Nadu 638476, INDIA.

Abstract- Most methods are post clustering measures of cluster validity i.e. they attempt to choose the best partition from a set of alternative partitions. In contrast, tendency assessment attempts to estimate the number of clusters before clustering occurs. In this paper, we investigate a new method called Intrusions pose a serious security risk in a network environment. Although systems can be hardened against many types of intrusions, often intrusions are successful making systems for detecting these intrusions critical to the security of these systems. New intrusion types, of which detection systems are unaware, are the most difficult to detect. Current signature based methods and learning algorithms which rely on labeled data to train, generally cannot detect these new intrusions. In addition, labeled training data in order to train misuse and anomaly detection systems is typically very expensive. We present a new type of clustering-based intrusion detection algorithm, unsupervised anomaly detection, which trains on unlabeled data in order to detect new intrusions. The selection of the number of clusters is an important and challenging issue in cluster analysis. A number of attempts have been made to estimate the number of clusters in a given data set. Trusted Precluster Count (TPCC) algorithm for automatically estimating the number of clusters in unlabeled data sets, which is based on an existing algorithm Dark Block Extraction (DBE) of a data set, using several common image and signal processing techniques. Our focus is to meddling detection in pre cluster count in unlabelled dataset using virtual assessment tendency.

Index Terms- Clustering, Intrusion, Virtual assessment Tendency, Trusted PreCluster Count, Dark Block Extraction

I. INTRODUCTION

Intrusion detection is “the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network”. Before data mining techniques are introduced into this field, intrusion detection was heavily dependent on a manually maintained knowledge base which contained signatures of all known attacks. Features of monitored network traffic were extracted and then compared with these attack signatures. Whenever a match was found, an intrusion was claimed to be detected and it was reported to the system administrator. Due to the difficulty and expense to manually maintain the knowledge base to reflect the ever-changing situations, it was not feasible to continue working in this traditional way. Recently, many researchers turned into data mining techniques to attack the problem. Clustering is a data mining technique used to place data elements into related groups without advance knowledge of the group definitions. It divides a set of objects (data points) into groups (clusters) such that the objects in a cluster are more similar to one another than to the objects in other clusters. The emerging data mining applications place many special requirements on clustering techniques, such as scalability with high dimensionality of data. A number of clustering algorithms have been developed over the decades in data base / data mining community. The selection of the number of clusters is an important and challenging issue in cluster analysis. A number of attempts have been made to estimate the number of clusters in a given data set. Most methods are post clustering measures of cluster validity, i.e., they attempt to choose the best partition from a set of alternative partitions. In contrast, tendency assessment attempts to estimate the number of clusters before clustering occurs. Our focus is on pre clustering tendency assessment, but for completeness, this project briefly summarizes some existing approaches to the post clustering cluster validity problem, before describing visual methods for cluster tendency assessment.

II. BACKGROUND

In data mining, clustering is the most important unsupervised learning process used to find the structures or patterns in a collection of unlabeled data. During the learning process, the similarities between pairs of data instances are considered and an optimized output is found which maximizes the similarities within the same group and the dissimilarities between different groups. While there is no absolute criterion to judge the optimality of the clustering result, the choice of similarity measures can surely lead to totally different groupings.

In many clustering algorithms, the spatial distance (usually Euclidean distance) between different instances is used to measure the similarity. This has proved to be successful in many applications. It is intuitive that similar objects are closer together while objects from different groups are far from each other. This intuition, however, is not always true in more complicated applications, especially when the shape of data clusters is not limited to be spherical. Figure 1 shows an example in which the popular k -means clustering algorithm cannot achieve satisfactory result.

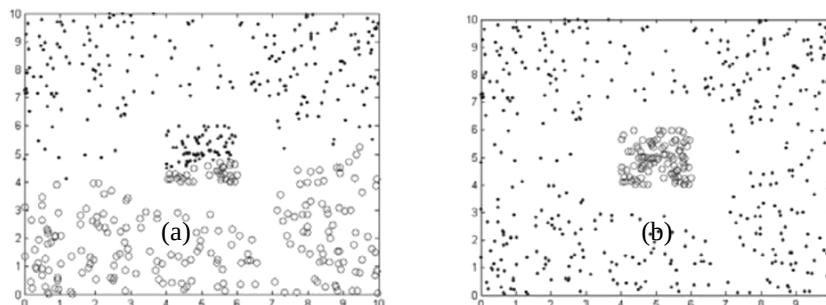


Figure 1. Clustering results on a dataset with 2 clusters
(a) with k -means (b) with FCC

To deal with more complex situations, it should be more suitable to take both local neighborhood and global information into consideration when deciding the membership of each data instance. In this paper, we propose a new clustering algorithm which uses fuzzy connectedness as the similarity metric.

In an image segmentation framework introduced, fuzzy affinities are assigned to the target data object during the classification. The affinity between a pair of pixels is defined as a weighted function of their adjacencies in the coordinate space, the intensity space, and the intensity gradient space. For a finite set of data points, the strength of the link from one point to another is defined as the reciprocal of the distance between them and it can be automatically defined based on statistical properties of the links within regions of points identified as belonging to the same groups. A chain is a sequence of ordered points and the strength of a chain is its weakest link. Taking fuzzy connectedness as the similarity metric, the task of a clustering process is to find the strongest path of every data point and assign the point to the group that path leads to.

Basic definitions that we use in this paper are presented below.

Definition 1: For a positive integer K , a K -FCC of a set D (of points) is a function which maps each point $c \in D$ into a

$(K+1)$ -dimensional vector $\sigma^c = (\sigma_0^c, \sigma_1^c, \sigma_2^c, \dots, \sigma_K^c)$, such that $\sigma_i^c \in [0,1]$, and for at least one k , $1 \leq k \leq K$, $\sigma_0^c = \sigma_k^c$;

for $1 \leq m \leq K$ and $m \neq k$, $\sigma_m^c = 0$ or $\sigma_m^c = \sigma_k^c$.

Definition 2: A fuzzy affinity on D is a function: $\phi : D^2 \rightarrow [0,1]$. Definition 3: A chain in $U \subseteq D$ from c^0 to c^N is a sequence $C^{(0,N)} \phi_U(C^{(0,N)}) = \min(\phi_U(c^{n-1}, c^n))$, $1 \leq n \leq N$.

(c, d) is the strength of the link (c, d) .

$(c^0, c^1, c^2, \dots, c^N)$ in U and the strength of the chain

Given the above definitions, a clustering process is to find the mapping of K -FCC which is deduced from the calculation of fuzzy affinities between pairs of points. In this process, the definition of the fuzzy affinity function ϕ is very essential; one can include the statistical information of each cluster to make the results more reliable. As shown in the example of Figure 1, when only the Euclidean distance between pairs of points is considered, it is not possible to separate the two clusters correctly. However, after we involve the statistical information of each cluster, e.g., the distribution of Euclidean distance among each point and its neighbors, the clustering result turns out to be perfect (Figure 1b). In the proposed method, in order to initialize the clustering process, we need a little prior knowledge. This is in the form of a few labeled data points in each possible cluster. This requirement, however, is usually easy to satisfy in intrusion detection applications. We discuss the details of our clustering algorithm in the next section.

III. METHODOLOGY

In order to calculate the fuzzy affinity of an instance c and another instance $d \in U_k$ (target cluster), besides their spatial adjacency, we also involve the statistical information about the cluster U_k . The distribution information of a certain variable (e.g., Euclidean distance between nearest neighbors) of U_k is stored and kept up to date. When a new instance comes, its probability of falling into U_k is calculated using the distribution information and the Euclidean distance between the new instance and its nearest neighbor d in U_k .

IV. IMPLEMENTING EXISTING REVATMETHODOLOGY

The Visual Assessment of (cluster) Tendency (VAT) [1] method readily displays cluster tendency for small data sets as grayscale images, but is too computationally costly for larger data sets. A revised version of VAT is presented here that can efficiently be applied to larger collections of data.

Initial Virtual Assessment Tendency (VAT) Algorithm

Input: An $n \times n$ scaled matrix of dissimilarities $D = [d_{ij}]$, with $1 \geq d_{ij} \geq 0$; $d_{ij} = d_{ji}$; $d_{ii} = 0$, for $1 \leq i, j \leq n$

Step (i): Set $I = \emptyset$, $J = \{1, 2, \dots, n\}$ and $P = (0, 0, \dots, 0)$.

Select $(i, j) \in \arg_{p \in J, q \in J} \max\{d_{pq}\}$.

Set $P(1) = i$, $I \leftarrow \{i\}$ and $J \leftarrow J - \{i\}$.

Step (ii): Repeat for $t = 2, \dots, n$

Select $(i, j) \in \arg_{p \in I, q \in J} \min\{d_{pq}\}$.

Set $P(t) = j$, update $I \leftarrow I \cup \{j\}$ and $J \leftarrow J - \{j\}$.

Step (iii): Form the re-ordered dissimilarity matrix $\tilde{D} = [\tilde{d}_{ij}] = [d_{P(i)P(j)}]$, for $1 \leq i, j \leq n$.

Output: A scaled gray-scale image $I(\tilde{D})$ so that $\max\{\tilde{d}_{ij}\}$ corresponds to white and $\min\{\tilde{d}_{ij}\}$ to black.

V. IMPLEMENTING BIGVAT METHODOLOGY

Assessment of clustering tendency is an important first step in cluster analysis. One tool for assessing cluster tendency is the Visual Assessment of Tendency (VAT) [1] algorithm. VAT produces an image matrix that can be used for visual assessment of cluster tendency in either relational or object data. However, VAT becomes intractable for large data sets. The revised VAT (reVAT) algorithm reduces the number of computations done by VAT, and replaces the image matrix with a set of profile graphs that are used for the visual assessment step. Thus, reVAT overcomes the large data set problem which encumbers VAT, but presents a new problem: interpretation of the set of reVAT profile graphs becomes very difficult when the number of clusters is large, or there is significant overlap between groups of objects in the data.

BigVAT [2] Algorithm steps include:

- (a) Choose segmented image from right hand side panel from the segmentation form.
- (b) Analyze the total number of pixel values and based on that pixel value, create a proxy controller which stores each pixel value of a bigVAT processing image
- (c) Total Proxy Value Count acts as the threshold value $= a$, from Selected Segmented image „M“
- (d) Considering bigVAT algorithm process, the `getpixel()` function is used to trace pixel value of 0th pixel proxy position and the pixel value will be allowed for dividend into „R“, „G“, „B“ values. Perform a distance transform to obtain a gray scale image and scale the pixel values.
- (e) Project the pixel values of the image on to the main diagonal axis to form a projection signal. Smooth the signal to obtain the filtered signal by an average filter.
- (f) the values will be formulated and the formulated value will be displayed in the corresponding list view position and the calculated pixel is replaced in the same position by using the function called `setpixel()`.
- (g) Once the entire process completed, the output will be displayed in the Screen.

VI. IMPLEMENTING DARK BLOCK EXTRACTIONMETHODOLOGY

One of the major problems in cluster analysis is the determination of the number of clusters in unlabeled data, which is a basic input for most clustering algorithms. In this paper, we investigate a new method called Dark Block Extraction (DBE) for automatically estimating the number of clusters in unlabeled data sets, which is based on an existing algorithm for Visual Assessment of Cluster Tendency (VAT) [1] of a data set, using several common image and signal processing techniques.

Dark Block Algorithm steps include:

Generate a VAT [1] image of an input dissimilarity matrix, Perform image segmentation on the VAT image to obtain binary image, followed by directional morphological filtering,

Applying a distance transform to the filtered binary image and projecting the pixel values onto the main diagonal axis of the image to form a projection signal

Smoothing the projection signal, computing its first-order derivative, and then detecting major peaks and valleys in the resulting signal to decide the number of clusters.

DBE method is nearly “automatic,” depending on just one easy-to-set parameter.

VII. IMPLEMENTING PROPOSED TRUSTED PRE CLUSTER COUNT METHODOLOGY

TPCC is an advanced method of detecting the number of clusters in a pre defined manner in order to give more accuracy to the segmented image. TPCC is a pre-clustering method, i.e., it does not require the data to be clustered, nor does it find clusters in the data. By using TPCC method, the segmented image is well clearly classified into pixel transformations by maintaining the entire pixel data in a proxy structure, i.e., in an array format. So the

calculations process is very little to find the density of the image in order to produce accuracy to the image. Trusted Pre Cluster Count Algorithm steps include:

- (a) Generate a VAT image.
- (b) Perform image segmentation on the VAT image.
- (c) Divide the image into matrix pixels as row „r“ X column „c“.
- (d) Create a proxy controller to store all matrix pixels values.
- (e) Calculate each pixel value and compare each pixel relate with neighbor pixel.
- (f) Group the similar result's pixels.
- (g) Calculate the number of groups separated which is equal to number of clusters.
- (h) The resulted cluster count will be the perfect pre cluster count value where it produces the VAT image into a super quality image.

VIII. EXPERIMENTAL RESULTS

In order to evaluate *FCC*, we tested the algorithm on a benchmark dataset, the network traffic data from the KDD Cup 1999 Dataset [4]. Two indicators, *Detection Rate* and *False Alarm Rate*, were used to measure the accuracy of the method. The *Detection Rate* shows the percentage of true intrusions that have been successfully detected. The *False Alarm Rate* is defined as the number of normal instances incorrectly labeled as intrusion divided by the total number of normal instances. A good method should provide a high *Detection Rate* together with a low *False Alarm Rate*.

Data preparation

The KDD dataset includes a wide variety of intrusions together with normal activities simulated in a military network environment. The simulated attacks fall in one of four major categories: DOS (denial of service), R2L (unauthorized access from remote machine), U2R (unauthorized access to local superuser privilege) and Probing (surveillance and other probing). Each instance in the dataset consists of the extracted features of a connection record. These features are either symbolic or continuous. In the *FCC* algorithm, the Euclidean distance is chosen as the metric to calculate the distance among different instances (points). Since the Euclidean distance can deal only with continuous types of data, all the symbolic features are removed from the data. At the same time, in order to avoid the dominance of any single (or several) feature(s) in the distance calculation, we normalize the data so that the values of every feature have the same range. In the experiments, the values of each feature are normalized with the minimum and maximum values of that feature so that they fall in the range of [0,1].

$$new_inst[i] = \frac{old_inst[i] - \min(feature_i)}{\max(feature_i) - \min(feature_i)} \quad (2)$$

Results

In our experiments, we consider a subset of the KDD data, which consists of 2776 instances. Except for a few seed points, the labeling information is not utilized in the experiments. It is, however, used after our experiments are performed, to evaluate the results and calculate the *Detection Rate* and *False Alarm Rate*.

In the first experiment, besides the normal instances, instances of four popular attacks are involved: smurf, ipsweep, neptune and back. For each type, only one seed point is labeled at the beginning, and four nearest neighbors are defined as the nearest neighborhood of each instance. As shown in Table 1, most attacks can be distinguished from the normal activities and the *Detection Rate* is as high as 97.8%. At the same time, approximately 3.6% normal activities are erroneously labeled as attacks. Among the instances labeled as attacks, about 95.8% of them are classified to the correct attack types. The difference between this accuracy and the *Detection Rate* is caused by the existing fuzziness among different intrusion types.

To test the ability of the *FCC* algorithm to detect new variations of attacks, we add instances of 10 other attack types, such as nmap, teardrop and dportsweep. While keeping the seed points the same as before, we repeat the experiment. Assuming that the new types of intrusion tend to have stronger fuzzy connectedness to instances of some existing types, we are able to keep the detection rate nearly the same as in the previous experiments. Nearly all the new attack instances are labeled with one of the four previous intrusion types and this keeps the *Detection Rate* at a high level. The actual *Detection Rate* is 98.1%, which is even better than before. At the same time, the *False Alarm Rate* stays at a rather low level.

Table 1. Experimental results of FCC

Experiment	Detection Rate	False Alarm Rate
A	97.8%	1.8%
B	98.1%	1.6%

The results of the second experiment clearly demonstrate the great potential of *FCC* in detecting newly emerging intrusions. It is feasible to apply *FCC* on currently available (unlabeled) data, and the clustering results can be used for future incremental learning. As long as the assumption that variations of intrusion types have similar

distributions to some known types holds, instances of new intrusion variations will have larger value of fuzzy connectedness to a certain cluster labeled as intrusion and be put into that cluster. Even though we may not know the actual type of the new variation, a correct intrusion alarm will be reported.

Parameters

In the FCC algorithm, two parameters are involved: (a) the number of seed points and (b) the number of neighbors in the neighborhood. The first parameter is decided using prior knowledge about the data. The minimum value is 1, which means that each cluster has at least one instance at the very beginning. Usually, the more seed points we have, the more information about the cluster will be given, and this will possibly provide more accurate results. However, depending on the distributions of different clusters, a small number of seed points may also give satisfactory results, as shown in Table 1.

The number of neighbors used to define the neighborhood is also important. As discussed earlier, when calculating the fuzzy connectedness of an instance to other instances in a certain cluster, both the local information about the distance between two instances and the statistical information about that cluster are involved. An instance x is fuzzy connected to another instance y only if y falls in the neighborhood of x . If the number of neighbors is too small, only a few instances tightly close to x are considered reachable from x , and this may separate x from the cluster it really belongs to; when this number is too large, the importance of local connectivity will be reduced and it is possible for two instances far from each other to be considered as fuzzy connected. This may bring two or more clusters close to each other forming a single cluster, which is also not desirable.

IX. CONCLUSION

In this paper we have concentrate on intrusion in pre cluster count in unlabelled dataset using VAT. Thus the cluster number (corresponding to the detection of major praks) extracted by DBE appears to be increasingly realiable. This is understandable since if increases, the segmented binary image will be less noisy, which is naturally helpful to subsequent processing. As long as the filter sizes are set to be less than the minimum meaningful cluster size, the larger they are, the more reliable the estimation of the cluster number should be. The emerging data mining applications place many special requirements on clustering techniques, such as scalability with high dimensionality of data. A number of clustering algorithms have been developed over the decades in data base / data mining community here in this paper intrusion are detected in the pre cluster count in order to overcome the traffic occurred in the network path. This may be helpful for various clustering formation using VAT algorithm.

X. REFERENCES

- [1] R. Bace and P. Mell, "Intrusion Detection Systems", *NIST Special Publications SP 800-31*, November, 2001.
- [2] Y. Guan, A. Ghorbani and N. Belacel. "Y-means: A Clustering Method for Intrusion Detection". Proceedings of Canadian Conference on Electrical and Computer Engineering. Montreal, Quebec, Canada. May 4-7, 2003.
- [3] G. Herman and B. Carvalho. "Multiseeded Segmentation Using Fuzzy Connectedness". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(5): 460-474, 2001.
- [4] KDD Cup 1999 data. University of California, Irvine. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>, 1999.
- [5] X. Li and N. Ye, "Decision tree classifiers for computer intrusion detection", *Real-time system security*, Nova Science Publishers, 2003.
- [6] M. Phung, Data Mining in Intrusion Detection? Intrusion Detection FAQ. October 24, 2000. http://www.sans.org/newlook/resources/IDFAQ/data_mining.htm
- [7] L. Portnoy, E. Eskin and S. Stolfo. "Intrusion detection with unlabeled data using clustering". Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). Philadelphia, PA: November 5-8, 2001.
- [8] A. Rosenfeld, "Fuzzy Digital Topology," *Information and Control*, vol. 40, pp. 76-87, 1979.
- [9] J. Ryan and M. Lin, "Intrusion Detection with Neural Networks", *Advances in Neural Information Processing Systems 10*, MIT Press, June, 1998.
- [10] J. Udupa and S. Samarasekera. Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation. *Graphical Models and Image Processing*, 58(3):246-261, 1996.
- [11] S. Zhong, T. Khoshgoftaar and N. Seliya, "Evaluating Clustering Techniques for Network Intrusion Detection", Proceeding of 10th ISSAT Int. Conf. on Reliability and Quality Design, pp. 149-155. Las Vegas, Nevada, USA. August 2004.