UDC 004.7.056.53

O. O. Didyk, O. V. Hrybovskyi

USING OF AGLOMERATIVE HIERARCHIC CLASTERIZATION TO ENSURE THE COMPUTER SYSTEMS SECURITY

Annotation. *Application of cluster analysis methods for the solving of problems of computer systems attacks detection is considered. The algorithm of agglomerative hierarchical clusterization is offered, allowing effectively solving a problem of allocation of dangerous areas of computer systems conditions.*

Key words: *clasterization, computer systems security, computer attacks detection.*

Introduction

Nowadays the importance of information security issues is obvious to all. Even a little thought is enough to understand these problem difficulties, with its roots traced from both the complexity and heterogeneity of modern information systems and of the need for a complex approach to security with the involvement of the legislative, administrative and program-technical measures.

Information systems can be secured in two ways. One of them is to prevent all the unauthorized access efforts (UAE), therefore creating completely secured system. But this is practically impossible via number of reasons:

- It is impossible to create absolutely secure system due to the errors in the software;

- Even the most secured system is vulnerable in front of the experienced people. A privileged user can simply violate security policies, and it can low the security level;

- The safer the system is, the harder is for users to work with it.

Therefore, if it is impossible to build a perfectly secure system, at least you need to discover all (or almost all) security policy violations and respond to them properly. This is the task that intrusion detection systems are designed to handle.

As the number and frequency of attacks are increasing all the time, it is very important to identify the attack at an early stage of their development and react to them on time. In critical case, the interception of the attack should be made much faster than human can react.

Another reason for the intrusion detection process automation is that attackers use automatic resources of distributed attacks realization. In this case thousands of server intrusion efforts can be registered from hundreds of locations within just a few hours, meaning several attacks per minute. In this case automatic detection system can help to trace attack source. If it's not present, the detection of attack and the person behind it is impossible.

General formulation of the computer attacks detection problem. The used model of attack detection task was proposed in [1, 2].

The action (or a sequence of actions) which was made by the offender and led to the security violation of the distributed information system (DIS), switching it from some safe into some dangerous condition, is considered as an *attack*. Attack always switches DIS from a safe state into a dangerous one. As the offender we mean person who actually makes such actions. *A normal behavior* is an action (the sequence of actions) that is not an attack.

The observer is either software or combined hardware and software device that has the ability to collect information about objects actions and about the resources status.

The intrusion detection task is the process of identifying the attack, which is based on the information about the DIS state received from the observer.

DIS resource status at the current time can be described by a certain set of parameters, which include both the characteristics of the resource utilization and information of the objects which are using the specific resource.

Time ordered sequence of DIS states is called DIS trajectory. *The trajectory of the object* is time ordered sequence of DIS resource states which are changing due to the influence exercised by this object to the DIS resource.

The trajectory of the object, which is carrying out impact on the DIS and resulting in a transition from a safe state into dangerous one, is called as an *attack trajectory*. Thus, a specific attack is considered as a trajectory in the N-dimensional parameter space. The set of all attacks can be divided into classes on a number of criteria.

For a certain class of attacks there may be more than just one path, there is a finite set of possible trajectories, forming *trajectories bundles* which are close each to other.

Because of less computational complexity, the discrete trajectory is used. Let the $\bar{t}$ be the ultimate duration of the given class attack and τ is the measuring time of the observed trajectory. Thus in the trajectory of the attack behavior there is $k = \lceil \bar{t} / \tau \rceil$ states. For each measurement of parameters (number of measurement is uniquely determined by k value) in parameter space there is a set of disconnected areas, so getting into one of these measurement areas means belonging trajectory L to the set on this measurement. These are "dangerous areas" that are, in fact, form a beam path sections attacks.

We believe that the attack is the activity, under the influence of which the system goes through a dangerous area on all the measurements. That is, if in the course of the session states k measurements at each measurement path fell into one of these areas, then the trajectory belongs to L - the set of trajectories corresponding to a certain class of attacks.

Thus, the task of detection of the specified class attacks is reduced to the task of:

1. Building a set of “dangerous areas” $G(t) = \{G_1(t), G_2(t), \ldots, G_l(t)\}$ for each measurement;

2. Defining the link between the session’s measured state and one of these areas $(x \in G)$.

Formulation of the problem. Thus, the solution to the problem of constructing a set of “dangerous areas” for each measurement reduces to the problem of clustering in the n-dimensional space. We have developed an agglomerative hierarchical clustering algorithm, which allows to effectively solving the problem of separating bunches of the attacks paths.

The agglomerative hierarchical clustering algorithm. The clustering is the process of finding clusters (groups) of objects that have a high similarity inside the cluster and low similarity between the clusters. In other words, the objects belonging to the same cluster more resemble each other than to the objects belonging to other clusters.

Unlike classification, clustering does not require predetermination of classes information, i.e. clustering is "learning via monitoring" unlike "learning via examples." In other words, clustering belongs to the "unsupervised learning" methods.

For clustering, a large number of algorithms and software have been developed [3, 4, 5, 6]. The reason for the diversity algorithm is that different applications use different data types - users need different clustering methods, which are adapted to the type of application and the type of clusters that are required.

The similarity of two objects is determined by a measure of similarity when grouping into clusters. While determining the similarity degree it's necessary to keep in mind that the attributes may have a different nature, i.e. they can be can be numeric, nominal, categorical, etc. Accordingly, it is necessary to apply different measures to such a various attributes.

The similarity between the objects is determined on the base of a *distance* between them, i.e. in the database containing such objects the similarity implies sameness of two or more such objects. In other words, the higher degree of similarity are two objects, must be less than the distance between them.

There is a number of approaches to determine the distance between two objects.

Existing data clustering algorithms can be divided into four types:

Methods based on decomposition;

Methods based on the density determination;

Lattice methods;

Hierarchical methods [6].

Agglomerative methods begin the clustering by considering each database object as a separate cluster. Next, the pair of clusters having the largest degree of similarity is combined into a single cluster. Such association is performed recursively until it reaches a predetermined threshold value. The threshold value may be set as the maximum number of clusters or the minimum distance between clusters.

Since hierarchical methods are quite simple and effective, no wonder they are the most popular. Based on these considerations, a hierarchical approach is chosen as a basis for building a clustering algorithm used to solve constructing a set of dangerous areas that are "bunches" of attacks on the trajectories of the computer system.

As it shown, the main purpose of clustering is to find such a set of clusters that objects belonging to the one cluster are as much similar to each other as possible, so the cluster will have minimal sparse. Another purpose is to find such a clustering in which objects in the different clusters are least similar to each other, i.e. clusters have the greatest *distance* between them. These criteria may be used to determine the *quality* of clustering.

Here is the definition measures of inter-cluster *distance*, the sparsity of the cluster and the clustering quality, as well as the clustering algorithm based on hierarchical agglomerative approach.

The most important measure is the clustering is *distance*. This measure determines how close or far are the individual clusters apart. Therefore, let's start by defining the distance between two clusters.

Let c_i and c_j to be the clusters in ***C*** clustering. The inter-cluster distance $d(c_i, c_j)$ between one-element clusters $c = \{\gamma\}$ and $c' = \{\gamma'\}$ is the distance between two objects γ and γ', so

$$d(c,c') = d_f(\gamma,\gamma'),$$

where d_f is the distance measurement between γ and γ' objects.

Let's review the situation where one or both cluster consists of a of two or more objects. Similarity of two clusters is determined by the similarity of the objects belonging to clusters.

The inter-cluster distance between two clusters *c* and *c'* is a *F* function of the paired distances between of objects when one of the objects belong to the cluster *c*, and the other – to the *c'* cluster, ie,

$$d(c,c') = F\left(\left\{d_f\left(\gamma_i,\gamma_j\right) \middle| \gamma_i \in c \wedge \gamma_j \in c'\right\}\right).$$

The measure the distance between the two single-element clusters is particular case of this distance measure, when there is only one pair of objects for comparison.

The *F* function, which determines the distance between two clusters can be defined in different ways [5]. Let's review the three functions which are used most often in the methods of agglomerative clustering.

The *minimum distance* function, which is the oldest and simplest measure of the clusters similarity, is defined as the distance between the closest members of two clusters:

$$d_{\min}(c,c') = \min\left(\left\{d_f(\gamma_i,\gamma_j) \middle| \gamma_i \in c \wedge \gamma_j \in c'\right\}\right).$$

Using this function, we can obtain clusters where each single object is more similar to the objects of its cluster than the objects in the other cluster. One of the problems arising when using this function is the tendency of forming stretched, or elongated, clusters, which can easily lead to significant differences between objects that are at opposite ends of the same cluster. [6] Another problem is that the clustering, which was done using this measure, is very sensitive to data noise.

The *maximum distance* function is directly opposite to the previous one. Here the inter-cluster distance between two clusters is defined as the distance between the objects that are most distant from each other:

$$d_{\max}(c,c') = \max\left(\left\{d_f\left(\gamma_i,\gamma_j\right)\middle|\gamma_i \in c \wedge \gamma_j \in c'\right\}\right).$$

This feature allows the creation of compact clusters that are not subjected to be well separated. However, the formation of stretched clusters using this feature is extremely difficult and, if the real object groups have an elongated shape, the resulting clusters may be inadequate. Another disadvantage, as in the case of the minimum distance function, is the high sensitivity to noise [6].

The third function determines the distance between two clusters as the average distance between all objects in pairs, i.e.

$$d_{avg}(c,c') = \frac{1}{|c|\cdot|c'|}\sum\nolimits_{\gamma_i \in c}\sum\nolimits_{\gamma_j \in c'} d_f(\gamma_i,\gamma_j)\,.$$

This feature is designed to find approximately spherical clusters.

Let’s define the clustering distance, which depends on the inter-cluster distance values of all pairs of clusters in clustering.

We are given a clustering $C = \{c_1, c_2, \ldots, c_n\}$, two clusters c_i and c_j in C clustering, the distance d (c_i, c_j) between clusters c_i and c_j. The clustering distance of C is a function of D, which can be defined as

$$C(D) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d(c_i,c_j)$$

the average inter-cluster distance of C clustering.

The further relative to each other clusters are positioned, the more clustering distance is and vice versa. Note that the clustering distance is a function from the clustering itself and not from the individual pairs of clusters.

Another important measure is the *average clustering sparseness*, which is a function of sparseness from all the clusters in the clustering. Let’s define the measure of the cluster discharge.

Let h_{med} to be the center of mass of the c cluster. Then the sparseness of the cluster is defined as

$$r(c) = \begin{cases} 1, & if\ |c| = 1 \\ \frac{1}{|c|}\sum_{i=1}^{|c|} d_f(h_{med}, \gamma_i), & if\ |c| > 1 \end{cases},$$

i.e. the average distance between members of cluster c and its center of mass.

As a consequence, the further apart the objects in the cluster are, the higher is its sparseness.

Given a clustering **C** = $\{c_1, c_2, \ldots, c_n\}$, two clusters c_i and c_j in clustering C, sparseness $r(c_i)$ in cluster c$_i$. The clustering sparseness C is the ***R*** function, defined as

$$\boldsymbol{R(C)} = \frac{1}{n}\sum_{i=1}^{n} r(c_i),$$

i.e. the average sparseness of all the clustering of clusters.

Sparseness clustering has high value if clustering includes sparse clustering and vice versa.

Clustering quality is a measure that describes how well the clustering was performed, i.e. how low is clusters sparseness and how far they are from each other.

Imagine we are given a clustering ***C*** = $\{c_1, c_2, \ldots, c_n\}$, clustering distance ***D(C)*** and the clustering sparseness ***R(C)***. The quality of clustering, which is a function of ***R(C)***, and ***D(C)***, defined as

$$\boldsymbol{Q(C)} = \frac{D(C)}{R(C)}.$$

When clustering the database it is required to find the best clustering matching to a predetermined quality function. This means that it is necessary to obtain the clustering with the greatest clustering distance and the lowest clustering sparseness.

Let’s imagine a general description of the clustering method based on the agglomerative hierarchical clustering algorithm (Fig. 1).

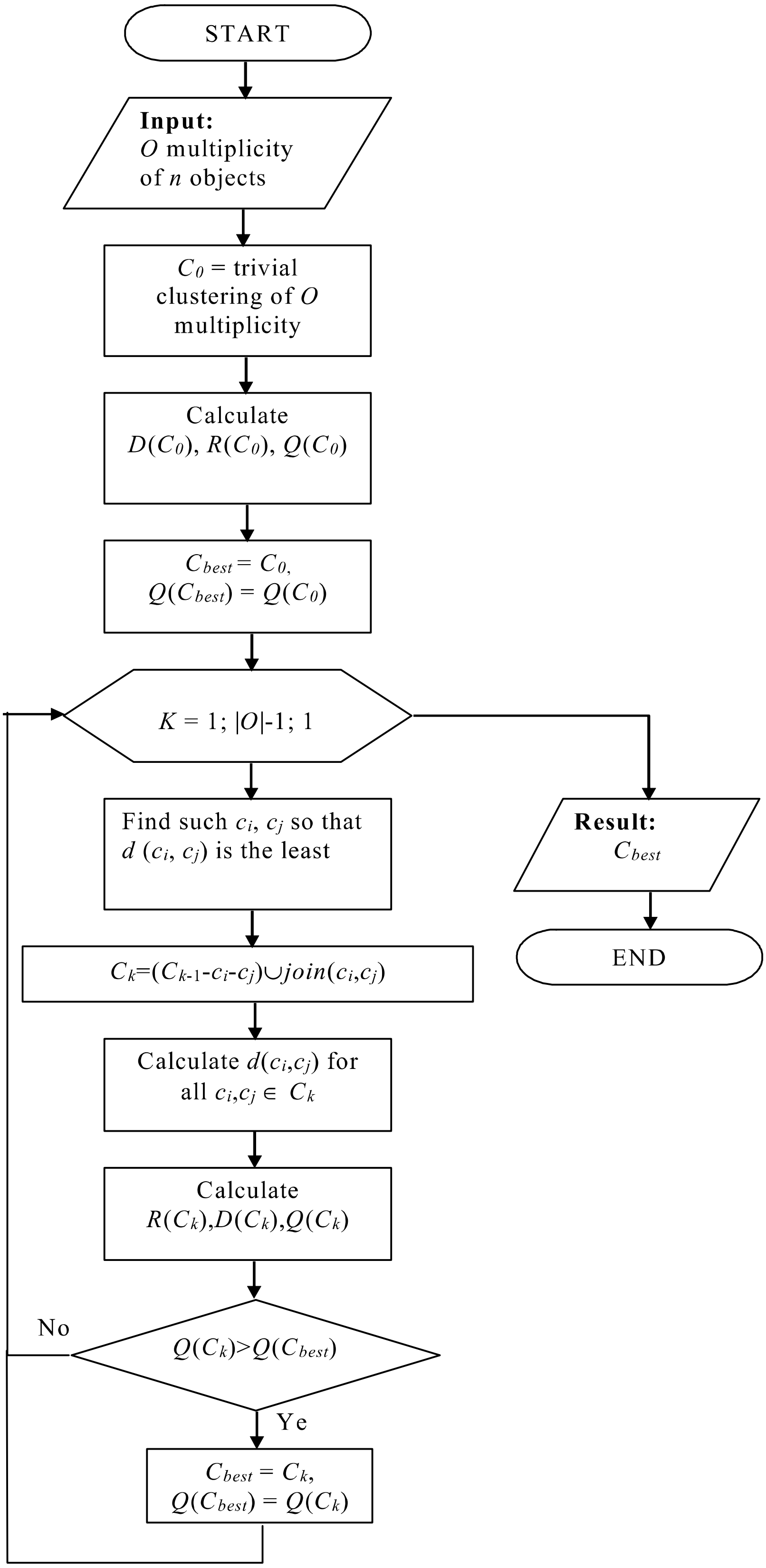

Fig. 1

In addition to the formation of clusters, the algorithm also calculates the sparseness of each cluster, as well as the distance and sparseness of the obtained clustering in general. The algorithm also determines which of the produced n clusterizations is the best, meaning which of them has the best quality indicator.

The algorithm takes as input parameter O set with n objects and generates n clustering of objects ranging from the trivial clustering C_0, consisting of a one-element cluster. The output is a C_{best}, clustering that is the best in terms of values of *clustering quality* measure. The application of quality measures eliminates the need to enter similarity threshold as an input parameter, which is required for most clustering algorithms. Determination of the threshold value is often difficult and requires an expert, which is often unacceptable conditions.

Moreover, the use of clustering quality measure provides the possibility of combining three different inter-cluster distance functions in one algorithm, which were shown above. As shown, each of these functions has its own advantages and disadvantages, and is suitable for clustering various types of clusters. At first, it is quite difficult to determine what type are the real clusters, that requires further expert assessment and a significant amount of preliminary experiments to determine the type of function used to determine the inter-cluster distance. Using clustering quality measure allows to do three clustering processes using stated inter-cluster distance functions, and to get a result of the three best clustering produced by these methods. As a result, using the degree of quality of each cluster, the best choice is made.

This approach gives a lot of flexibility in terms of functions choice to determine the inter-cluster distance, as well as the need to deprive the necessity to arrange the interactive communication with an expert in the process of clustering to determine the similarity threshold value and assess the quality of the obtained clustering.

Conclusions

For practical algorithm testing, we have used the data set for the evaluation of intrusion detection systems of Department of perspective research programs (DARPA) of the US Department of Defense [7]. The end results showed the efficiency of the developed agglomerative hierarchical clustering algorithm to solve the problem of constructing sets of dangerous areas of the state of the computer system.

References:

1. Гамаюнов Д.Ю., Качалин А.И. Обнаружение атак на основе анализа переходов состояний распределенной системы // Искусственный интеллект. 2004. – № 2. – С.49-53.
2. Промежуточный научно-технический отчет по первому этапу НИР «Невод». – М.: ф-т ВМК МГУ, 2004.
3. P.K. Garg, S. Bhansali. Process programming by hindsight. In Proceedings of the 14th International Conference on Software Engineering, p. 280-293. IEEE Computer Society Press, May 1992.
4. Davis R., 1982. TEIRESIAS: Application of meta-level knowledge // Knowledge-based systems in Artificial Intelligence. N.Y.: McGraw-Hill.
5. W. Shen, K. Ong, B. Mitbander, C. Zaniolo. Metaqueries for data mining. In Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996, pp. 375-398.
6. W. Ziarko, N. Shan. A Rough Set-Based Method for Computing All Minimal Deterministic Rules on Attribute-Value Systems, Technical Report CS-93-02 dept. of Computer Science, University of Regina, Canada, 1993.
7. http://www.ll.mit.edu/IST/ideval/data/data_index.html