An Efficient Hybrid Clustering-PSO Algorithm for Anomaly Intrusion Detection

Hongying Zheng
College of Computer Science and Engineering, Chongqing University, Chongqing 400044, China
zhenghongy@cqu.edu.cn

Meiju Hou, Yu Wang
College of Computer Science and Engineering, Chongqing University, Chongqing 400044, China
meijuhou0119@126.com，chuang_bei@sohu.com

***Abstract*—Generally speaking, in anomaly intrusion detection, modeling the normal behavior of activities performed by a user or a program is an important issue. Currently most machine-learning algorithms which are widely used to establish user's normal behaviors need labeled data for training first, so they are computational expensive and sometimes misled by artificial data. This study proposes a PSO-based optimized clustering method IDCPSO for modeling the normal patterns of a user's activities which combines an unsupervised clustering algorithm with the PSO technique, PSO algorithm is used to optimize the clustering results and obtain the optimal detection result. IDCPSO needs unlabeled data for training and automatically establishes clusters so as to detect intruders by labeling normal and abnormal groups. The famous KDD Cup 1999 dataset is used to evaluate the proposed system. In addition, we compare the performance of PSO optimization process with GA.**

***Index Terms*—PSO, Unsupervised Clustering, Anomaly Intrusion Detection, Optimization**

I. INTRODUCTION

Currently, Due to the advance of computer and communication technology and the reliance on Internet and world wide connectivity, damages caused by unexpected intrusions and crimes related to computer systems have been increased rapidly; A computer system should provide confidentiality, integrity and availability against denial of service; therefore, it is very important that the security mechanisms of a system are designed so as to prevent unauthorized access to system resources and data. Firewalls are hardware or software systems placed in between two or more computer networks to stop the committed attacks by isolating these networks using the rules and policies determined for them. However, it is very clear that firewalls are not enough to secure a network completely because the attacks committed from outside of the network are stopped whereas inside attacks are not. This is the situation where intrusions detection systems (IDSs) are in charge. IDSs are used in order to stop attacks, recover from them with the minimum loss or analyze the security problems so that they are not repeated, the performed tasks of an IDS include [1]: (1) Monitoring and analyzing user and system activities;(2) Audit of system structure and fault;(3) Recognition activity model mapping known attacks and alert;(4) Statistic analysis of abnormal behavior model;(5) Evaluating the integrity of systems and data files. Intrusion detection systems assume that they can detect an intruder by examining such parameters as network traffic, CPU and I/O utilization, user location, and file activity for signs of an attack.

In order to detect intruders, a number of detection methods have been proposed, these methods are mostly based on Denning's intrusion detection model. In this model, audit records, network packets and any other observable activity service as the basis for detecting abnormalities in the system. These techniques can be categorized into anomaly detection and misuse detection. Anomaly detection systems flag observed activities that deviate significantly from the established normal usage patterns as anomalies (i.e. possible intrusion). Therefore, the main issues in anomaly detection systems thus become the selection of threshold levels so that neither of the false negative rate and the false positive rate is unreasonably magnified. The main advantage of anomaly detection approaches is the ability to detect novel attacks or unknown attacks against systems, variants of known attacks, and deviation of normal usage of programs. And the disadvantage of anomaly detection approaches is that if the attack fits the established profile of the user, it is often difficult to detect it; another drawback is that a malicious user can train the anomaly detection system to learn the attack's malicious behaviors as normal by changing the profile slowly over time. Finally, anomaly intrusion detection methods always result in high false positive rate. While misuse detection systems, use patterns of well-known attacks or weak spots of the system to match and identify known intrusion patterns or signatures. The main issues in misuse detection systems are how to write signatures or patterns that encompass all possible variations of the attacks. The main advantage of misuse detection approaches is that they have high detection rare. Whereas, Novel attacks or unknown attacks or even

doi:10.4304/jsw.6.12.2350-2360

variants of common attacks often go undetected.

Unsupervised anomaly detection methods have been addressed recently, these methods take a set of unlabeled data as input and attempt to find intrusion buried within the data. To detect a new attack, they do not need any prior knowledge about training data and new attacks. Clustering is the unsupervised classification of input items into groups (clusters) without any prior knowledge. It is promising to detect unknown attacks in intrusion detection automatically. Furthermore, the clusters data can be analyzed for more information, i.e. the signature of new attacks. Most unsupervised anomaly detection methods are based on two basic assumptions about data. First, the number of normal instances vastly outnumbers that of anomalies. Second, data instances of the same classification (type of attack or normal) should be close to each other in the feature space under some reasonable metrics, and instances of different classifications are far apart. For example, Portnoy and Eskin presented a clustering-based anomaly detection approach in Ref. [2] which used the labeled clusters to classify network data according to the label of the nearest cluster. To overcome the drawback of a finite audit data set indicated the static activity of a user, authors in Ref. [3] proposed an anomaly intrusion detection method that continuously modeled the normal behavior of a user over the audit data stream, clusters of feature values corresponding to activities observed thus far in an audit data stream were identified by a statistical grid-based clustering algorithm for a data stream. In some cases, the clustering method is used to provide other algorithms with higher-qualified training items before training. For example, authors in Ref. [4] proposed an SVM-based intrusion detection system, which combined a hierarchical clustering algorithm and the SVM technique. The hierarchical clustering algorithm provided the SVM with fewer, abstracted, and higher-qualified training instances that were derived from the KDD Cup 1999 training set. It was able to greatly shorten the training time, but also improve the performance of resultant SVM. In Ref. [5], a new RFID intrusion detection method that was based on fuzzy c-Means clustering was proposed, the new RFID intrusion detection method can enhance the security and speed up the intrusion detection of RFID systems. In Ref. [6], the author proposed a new algorithm based on time stamped hierarchical clustering for intrusion detection. It incrementally clustered incoming data objects of normal behavior, and produced a precise model. Using each clusters time stamp, the algorithm can dynamically remove some expired clusters from the model, and also can produce some new cluster, which made clustering method more suitable for real network environment, the algorithm was less sensitive to noise data objects, and had low computer resource consumptions. In Ref. [7], the author investigated a hybrid clustering based filtering approach for high dimensional data clustering in detecting anomaly based network intrusions. A hierarchical conceptual clustering algorithm (COBWEB) had been used for data filtering and Farthest First Traversal (FFT) clustering technique for classification of rare attacks, the proposed approach was quite effective in comparison to their individual counterparts in detecting network intrusions, especially that come under U2R and R2L rare attacks category.

Although many kinds of clustering methods, such as FCM, K-MEANS, are widely used in intrusion detection, few clustering algorithms guarantee a global optimal solution. Therefore, to find global optimal clusters instead of local optimal results, this article presents a new approach for network anomaly intrusion detection called IDCPSO (Clustering and PSO). In the proposed method, the authors model the normal behaviors of a user or a program by using clustering the training data set and the optimal clustering results are obtained by means of PSO algorithm. PSO has many advantages over other evolutionary computation techniques (for example, genetic algorithm (GA)) such as simple implementation, faster convergence rate and fewer parameters to adjust. PSO is presented so as to combine it with clustering techniques for finding the minimum of the fitness function, producing a good result and enhancing the detection rate of intrusion. IDCPSO can combine the advantages of both PSO with clustering. Results are also compared with genetic algorithm. The procedure of intrusion detection is composed of the three parts, that is, (1)modeling the normal behaviors of a user or a program by creating clusters from unlabeled training datasets; (2) labeling clusters as ‘normal’ or ‘anomalous’; (3) using the labeled clusters to classify network data.

The remainder of this paper is organized as follows. Section 2 discusses the related work of anomaly intrusion detection based on supervised and unsupervised learning. Section 3 describes the proposed method of IDCPSO in detail. Section 4 shows the experiment results on KDD cup. Section 5 gives some conclusions.

II. THE RELATED WORK

In general, depending on whether the class labels are provided for learning, anomaly intrusion detection algorithms can be classified as either supervised or unsupervised. The supervised algorithms mostly exhibit excellent classification accuracy on the data with known attacks, but the accuracy of supervised algorithms deteriorates significantly if unknown attacks are present in the test data. While the unsupervised algorithms exhibit no significant difference in performance between known and unknown attacks, in other words, the performance of unsupervised learning is not affected by unknown attacks [8].

A. Supervised Anomaly Detection

Many supervise anomaly detection methods were proposed, for example k-Nearest Neighbor, SVM, MLP, and Decision Trees and so on, they are described in detail as follows.

The k-Nearest Neighbor is a classical algorithm that

finds k examples in training data that are closest to the test example and assigns the most frequent label among these examples to the new example. The only free parameter is the size k of the neighborhood. In Ref. [9], the task of the nearest neighbor approach was to remove noisy data and to build the set of original clusters. The aim of clustering by nearest neighbor algorithm was to reduce the size of data sets to a moderate one suitable for genetic algorithms at the second stage and to reduce the computing time as much as possible. In Ref. [10], the author used kNN classifier which was also a popular method in text categorization. The kNN classifier was used to classify program behavior as normal or intrusive. Program behavior was represented by frequencies of system calls. Each system call was treated as a word and the collection of system calls over each program execution as a document. This method seemed to offer some computational advantages over those that seek to characterize program behavior with short sequences of system calls and generated individual program profiles. In Ref. [11], the authors proposed a hybrid learning model based on the triangle area based nearest neighbors (TANN) in order to detect attacks more effectively. In TANN, the k-means clustering was firstly used to obtain class centers corresponding to the attack classes, respectively. Then, the triangle area by two class centers with one data from the given data set was calculated and formed a new feature signature of the data. Finally, the k-NN classifier was used to classify similar attacks based on the new feature represented by triangle areas.

Some supervised algorithms generalize equally well to the data with unknown attacks, for example SVM, which can be attributed to the fact that the free parameters of this algorithm are motivated by learning-theoretic arguments aimed at maintaining an ability to generalize to unseen data. In Ref. [12], the author applied the ANN and SVM algorithms to employ a frequency-based encoding method Instead of using a conventional sequence-based encoding scheme for intrusion detection. SVM is a powerful tool for classification problems, but still has some drawbacks. The first problem is that SVM is sensitive to outliers or noises. Second, SVM is designed for the two-class problem, so it has to be extended for multiclass problems. Some methods to improve the performance of SVM were proposed. In Ref. [13], the author projected input data into a high dimensional space by using the discriminant vectors extracted by KFDA. Then they constructed the optimal decision tree for multiclass SVM, based on the results of fuzzy clustering on the projected data. In Ref. [14], an improved incremental SVM algorithm, named RS-ISVM, was developed. To reduce the noise generated by feature differences, the author proposed a modified kernel function U-RBF, with the mean and mean square difference values of feature attributes embedded in kernel function RBF. Moreover, in order to shorten the training time, a concentric circle method was suggested to be used in selecting samples to form the reserved set.

Training of a multi-layer perception involves optimizing the weights for the activation function of neurons organized in network architecture. The free parameter is the number of hidden neurons. In Ref. [15], a hybrid MLP/CNN neural network was constructed in order to improve the detection rate of time-delayed attacks, the proposed approach can detect time-delayed attacks efficiently with chaotic neuron and this approach also exhibited a lower false alarm rate when detects novel attacks.

Decision trees build classification models based on recursive partitioning of data. Typically, a decision tree algorithm begins with the entire set of data, splits the data into two or more subsets based on the values of one or more attributes, and then repeatedly splits each subset into finer subsets until the size of each subset reaches an appropriate level. The entire modeling process can be represented in a tree structure, and the model generated can be summarized as a set of "if-then" rules. Decision trees are easy to interpret, computationally inexpensive, and capable of coping with noisy data. Therefore, the techniques have been widely used in intrusion detection. In Ref. [16], SURPASS was proposed which was highly effective in handling large data. SURPASS incorporates linear discriminants into decision trees' recursive partitioning process. In SURPASS, the information required to build a decision tree was summarized into a set of sufficient statistics. By reading a subset of the data from storage space to main memory one at a time, the data size that can be handled by this algorithm was independent of memory size.

B. Unsupervised Anomaly Detection

While in unsupervised learning, the data are not labeled. From the perspective of machine learning, the searching for clusters is unsupervised learning. To perform clustering is to try to discover the inner nature of the data structure as a whole, and to divide the data into groups of similarity. For example, in Ref. [17], The RT-UNNID system was introduced to be capable of intelligent real-time intrusion detection using unsupervised neural networks. Unsupervised neural nets can improve their analysis of new data over time without retraining. The aim of using unsupervised neural nets was to detect known and new attacks in network traffic. Some researchers use SOMs [18, 19] to learn patterns of normal system activities in anomaly detection tasks and employ a three-layer SOM to detect anomalous user behavior and anomalous network traffic.

There are generally three types of clustering algorithms that is Partition-based clustering, Hierarchical clustering and Density based clustering.

Partition-based clustering divides data sets into non-overlapping clusters. Given a predefined number of clusters, find the optimal partition for each point. The k-means algorithm is a well-known example of this kind of clustering methods. In Ref. [20], based on developed HSMM, an algorithm of anomaly detection was presented, which computed the distance between the

processes monitored by intrusion detection system and the perfect normal processes. To improve accuracy, the segmental K-means algorithm was applied as training algorithm for the hidden semi-Markov model. In Ref. [25], the author proposed a novel algorithm (IDCPSO) for modeling the normal behavior of user activities. There are two stages in the implementation of the algorithm. The first stage is to cluster network data and the second is the optimization process. Compared to genetic algorithms, this paper demonstrated higher DR and lower FPR and FNR.

While hierarchical clustering builds a cluster hierarchy. The hierarchy is a tree of clusters. Every node in the tree contains child clusters while sibling clusters share a common parent node. Typical hierarchical clustering algorithms are BIRCH and CURE. In BIRCH, a CF (Clustering Feature) tree which was used to summarize cluster representations was generated dynamically. After the CF tree was built, any clustering algorithm such as a typical partitioning algorithm was then used. For example, in Ref. [4], BIRCH clustering algorithm was first used to produce a reduced and high quality dataset from the original KDD Cup 1999 dataset before SVM training. The proposed system constructed four CF trees for DoS, U2R, R2L, and Probe attacks, respectively, and one CF tree for normal packets. The proposed system could reach high detection accuracy with a low false positive rate.

Unlike other methods, the density based clustering method regards a cluster as a region in a data space with the proper density of data objects. Typical density-based clustering algorithms are DBSCAN and CLIQUE. In Ref. [21], the author proposed an anomaly detection method, which utilized a density-based clustering algorithm DBSCAN for modeling the normal behavior of a user's activities in a host. The common knowledge of activities in the transactions of a user was represented by the occurrence frequency of similar activities by the unit of a transaction as well as the repetitive ratio of similar activities in each transaction. In Ref. [22], a new density-based and grid-based clustering algorithm that is suitable for unsupervised anomaly detection was presented. The system can be trained with unlabelled data and was capable of detecting previously unseen attacks.

C. Hybrid Anomaly Detection

In related work, a hybrid machine learning model based on combining the unsupervised and supervised classification techniques was proposed. One clustering technique is used as the first component for "pre-classification" and one supervised classification technique as the second component for the final classification task. For example, Latifur Khan et al. [23] presented a new approach that is combination of Support Vector Machines and Dynamically Growing Self-Organizing Tree (DGSOT) algorithm. SVM was used for classification and hierarchical clustering for analysis. Clustering analysis helped find the boundary points, which were the most qualified data points to train SVM, between two classes. The method started with an initial training set and expanded it gradually using the clustering structure produced by the DGSOT algorithm. In Ref. [24], Cheng proposed a multiple-level hybrid classifier which combines the supervised tree classifiers and unsupervised Bayesian clustering to detect intrusions, this new approach had high detection and low false alarm rates.

III. THE IDCPSO ALGORITHM

The IDCPSO algorithm consists of two stages. They are stated as follows:

1) The stage of clustering: To establish the set of original clusters using the clustering method by grouping very similar instances into a cluster and filter noisy data objects based on some similarity or dissimilarity metrics.

2) The stage of PSO optimization: To optimize original clusters by PSO algorithms and obtain the near optimal result, then label the cluster including most activities as the normal according to above assumptions.

A. Unsupervised Clustering

Clustering analysis is a method for grouping and classifying objects without category labels. Let the set of n points $\{ c_1, c_2, \ldots, c_n \}$ be represented by the data set Q and the number of clusters be represented by K, In Ref. [26], clustering is described as an assignment problem. Each cluster has a unique cluster label in $(1,...,K)$; the vector c assigns a cluster label $c_i \in (1,...,K)$ to the i-th point. Fig.1 shows an assignment instance. the number of data item is equal to 5, namely |Q|=5, K=2 and the assignment vector c=(2,2,1,2,1). The vector c shows that the data set is divided into two clusters, the first cluster is composed of data items 3, 5 and the other cluster is data items 1, 2, 4.

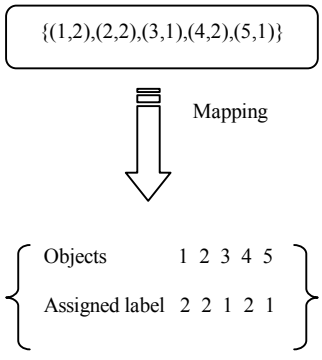

Figure 1. The assignment instance

For an assignment, we can evaluate the clustering quality by clustering criterion. For example, squared error based on cluster mean in (1) which is widely used in k-means. m_k denotes the vector of cluster center of the k-th cluster and $\|\bullet\|$ is Euclidean distance. Besides this, other clustering criterions are taken into account.

Such as (2) and (3) which define the total of inter-cluster distance and intra-cluster distance separately.

$$J = \sum_{k=1}^{K} \sum_{c_i=k} \left\| c_i - m_k \right\| \quad (1)$$

$$J = \sum_{k=1}^{K} \sum_{\substack{c_i=k \\ c_j=k}} \left\| c_i - c_j \right\| \quad (2)$$

$$J = \sum_{k=1}^{K} \sum_{\substack{c_i=k \\ c_j \neq k}} \left\| c_i - c_j \right\| \quad (3)$$

For a given data set, $\left(c : \forall i \in (1,...,N), c_i \in (1,...,K)\right)$ is the set of all feasible clustering, we should find the optimal solution of c about clustering criterion to enhance clustering quality. Therefore, clustering is converted into the optimizing problem described as the following (4) and (5).

$$\text{Minimize } J(c) \quad (4)$$

$$\text{Subject to } c \in C \quad (5)$$

B. The Stage of PSO Optimization

After above steps, the set of initial clusters can be obtained. This stage is a PSO optimization process at which initial clusters are used to set up the initial particle. The PSO optimizes a fitness function by iteratively improving a swarm of solution vectors, called particles, based on special management of memory. Each particle is modified by referring to the memory of individual and swarm's best information. Due to the collective intelligence of these particles, the swarm is able to repeatedly improve its best observed solution and converges to an optimum. This will be described in the following.

1) Framework of PSO

The PSO method was developed by Kennedy and Eberhart [27, 28] which has been successfully applied to many science and practical fields [29, 30, 31, 32, 33, 34]. PSO is a sociologically inspired population based optimization algorithm. Each particle is an individual, and the swarm is composed of particles. In PSO, the solution space of the problem is formulated as a search space. Each position in the search space is a correlated solution of the problem. Particles cooperate to find the best position (best solution) in the search space (solution space). Each particle moves according to its velocity. During iteration, the particle movement is computed as follows:

$$x_i(t+1) \leftarrow x_i(t) + v_i(t) \quad (6)$$

$$v_i(t+1) \leftarrow w v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (gbest(t) - x_i(t)) \quad (7)$$

In (6), (7), $x_i(t)$ is the position of particle i at time t, $v_i(t)$ is the velocity of particle i at time t, $pbest_i(t)$ is the best position found by particle i itself so far, $gbest(t)$ is the best position found by the whole swarm so far, w is an inertia weight scaling the previous time step velocity, c_1 and c_2 are two acceleration coefficients that scale the influence of the best personal position of the particle ($pbest_i(t)$) and the best global position ($gbest(t)$), which are popularly chosen to be $c_1 = c_2 = 2$, r1 and r2 are random variables between 0 and 1. The process of PSO is shown as Fig.2.

```
Initialize a population of particles with random positions and velocities in
the search space.
While (termination conditions are not met)
{
  For each particle do i
  Update the position of particle i according to (6).
  Update the velocity of particle i according to (7).
    Map the position of particle i in the solution space and evaluate its
      fitness value according to the fitness function.
  Update pbest_i(t) and gbest(t) if necessary.
  End for
}
```

Figure 2. The process of the PSO algorithm

2) Particle representation and initial swarm generation

We let each particle represent a decision for cluster assignment using a vector of N elements and each element is binary value. N is the number of data items in data set Q, namely |Q|=N. Fig.3 shows an illustrative example for the ith particle which corresponds to a cluster assignment that the data items 1,2,4 are belong to a cluster and the items 3,5 are the other cluster .

The PSO randomly generates an initial swarm of M particles, where M is the swarm size. These particle vectors will be iteratively modified based on collective experiences in order to improve their solution quality.

	1	2	3	4	5
The ith particle:	1	1	0	1	0

Figure 3. An example for the ith particle representation

3) Fitness function

The fitness function represents the goodness degree of a solution. In IDCPSO, we take into account not only intra-cluster distance but also inter-cluster distance. The intra-cluster distance indicates the degree of apart from; the larger the value, the farther distance the two clusters. Let $d(X,Y)$ be the intra-cluster distance. While the inter-cluster distance shows the degree of similarity. The less the distance, the more similarity the two data items in the same cluster. Let x_i, y_i be the input data items, m_i is the cluster center equaling to the mean of data items which belong to the same cluster. Therefore, the

fitness function of the particle vector can finally be defined as (8).

$$J' = \left(J_X + J_Y\right) \Big/ d_{(X,Y)} \tag{8}$$

$$J_X = \sum_{i=1}^{N}\sum_{j=1}^{N}\left\|x_i, x_j\right\| \tag{9}$$

$$d_{(X,Y)} = \left\|m_X, m_Y\right\| \tag{10}$$

$$m_X = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{11}$$

4) Particle vector modification

In IDCPSO, since the particle position is a binary string, the particle position vector modification is completed using (12), where $Sig(x) = 1/(1+\exp(-x))$. The parameter r_3 is uniformly distributed random numbers in [0, 1]. The particle flies through potential solutions toward pbesti and gbest in a navigated way while still exploring new areas by the stochastic mechanism to escape from local optimum [29].

$$x_{id}(t+1) = \begin{cases} 1, r_3 < Sig(v_{id}(t+1)) \\ 0, r_3 \geq Sig(v_{id}(t+1)) \end{cases} \tag{12}$$

5) Stop criterion

The IDCPSO will be terminated if the max number of iterations is met or the fitness function does not change any more in a given number (the algorithm converges to an optimum).

C. Labeling Clusters

Let an assumption that normal data items constituting an overwhelmingly large portion of the training dataset be satisfied. We therefore label some percentage N of the clusters containing the largest number of data items associated with them as 'normal', the rest of the clusters are labeled as 'anomalous'. Besides this, there is another problem that should been taken into consideration, that is, there may be many different kinds of normal network activity, this, in turn, might produce a large number of such 'normal' clusters which will have a relatively small number of data items. Therefore, for labeling a cluster, we should calculate the number of data items as well as consider the distance from other clusters. If the number of data items in a cluster is lowest and the distance from the other clusters is largest, we labeled it ‘anomalous’.

D. Anomaly Detection Method

Once the clusters are created from a training data set, the system is ready to perform detection of intrusions. Given a data item d, we should standardize d to d’ and find a cluster which is closest to d’ under the Euclidean distance, i.e. a cluster c in the cluster set C, such that for all C-c, dist (d’, c) ≤ dist (d’, C-c) .Assign type of cluster c (either normal or anomalous) to data item d.

IV. EXPERIMENTAL RESULTS

In this section, we describe the experiments conducted to evaluate the proposed system. The proposed system is tested on a Celeron processor 1.5 GHz with 512 RAM running Windows XP and coded by matlab 6.5.

A. Data Source

The KDD Cup99 dataset [35] is derived in 1999 from the DARPA98 network traffic dataset by assembling individual TCP packets into TCP connections. It is the benchmark dataset used in the International Knowledge Discovery and Data Mining Tools Competition, and also the most popular dataset that has ever been used in the intrusion detection field. Each TCP connection has 41 features with a label which specifies the status of a connection as either being normal, or a specific attack type. There are 38 numeric features and 3 symbolic features falling into the following four categories:

1) Basic features: 9 basic features are used to describe each individual TCP connection.

2) Content features: 13 domain knowledge related features are used to indicate suspicious behavior having no sequential patterns in the network traffic.

3) Time-based traffic features: 9 features are used to summarize the connections in the past 2 s that have the same destination host or the same service as the current connection.

4) Host-based traffic features: 10 features are constructed using a window of 100 connections to the same host instead of a time window.

Table I shows the basic features of KDD cup99. Because of the limitation of searching space and time, dimensionality should be reduced. finally, only 7 features are chosen. They are src_bytes, dst_bytes, count, srv_count, dst_host_count, dst_host_srv_count, dst_host_same_src_port_rate.

B. Data Preparing

The dataset contains about five million connection records as training data and about two million connection records as test data. Attacks fall into four categories: (1) Denial of Service (DoS): making some computing or memory resources too busy to accept legitimate users access these resources. (2) Probe (PRB): host and port scans to gather information or find known vulnerabilities. (3) Remote to Local (R2L): unauthorized access from a remote machine in order to exploit machine’s vulnerabilities. (4) User to Root (U2R): unauthorized access to local super user (root) privileges using system’s susceptibility.

In order to reduce the size of the dataset, we randomly select 952 records which satisfy the above assumption. Table II shows detailed information about the number of all records. It is important to note that the test data includes specific attack types not present in the training

data. This makes the intrusion detection task more realistic.

TABLE I
THE BASIC FEATURE NAMES

Feature name	Description	Type
Duration	length (number of seconds) of the connection	Count
Protocol_type	type of the protocol, e.g. tcp, udp, etc.	String
Service	network service on the destination, e.g., http, telnet	String
Src_bytes	number of data bytes from source to destination	Count
Dst_bytes	number of data bytes from destination to source	Count
Flag	normal or error status of the connection	String
Land	1 if connection is from/to the same host/port; 0 otherwise	Boolean
Wrong_fragment	number of ``wrong'' fragments	Count
Urgent	number of urgent packets	Count

TABLE II
NUMBER AND DISTRIBUTION OF TRAINING AND TESING DATASET

Connection type	Training dataset	Testing dataset
Normal	633(66.49%)	2000(44.56%)
DoS	96(10.08%)	350(7.8%)
PRB	65(6.83%)	560(12.48%)
R2L	73(7.67%)	678(15.11%)
U2R	85(8.93)	900(20.05%)

TABLE III.
DESCRIPTIVE STATISTICS OF THE 952CASES OF THE DATASET IN KDD CUP99

Feature name	Type	m_j (1.0e+003)	s_j (1.0e+003)
src_bytes	Continuous	0.3808	0.1601
dst_bytes	Continuous	2.0245	7.8162
count	Continuous	0.2782	0.2467
srv_count	Continuous	0.2733	0.2507
dst_host_count	Continuous	0.1935	0.0959
dst_host_srv_count	Continuous	0.2388	0.0588
dst_host_same_ src_port_rate	Continuous	0.0006	0.0005

C. Data Normalization Processing

We encounter a problem when processing instances whose different features are on different scales. This will cause bias toward some features over other ones. To solve this problem, these raw data sets need to be normalized. 41 features can be divided into 4 categories (‘Boolean’, ’String’, ’Count’, ’Rate’). Category ‘Boolean’ is 0 for ‘no’ and 1 for ‘yes’; category ‘Count’ is normalized according to (13); category ‘Rate’ remains unchanged. Category ‘String’ is mainly used to analyze features of clusters. m_j is the average feature instance of j-th feature and s_j is the standard deviation. Let the input data set be $I = \{I_{ij} \mid i = 1,...,N, j = 1,...,P\}$, where N is the number of total sessions and P is the number of features. Table III gives statistic results about m_j and s_j.

$$I'_{ij} = \frac{I_{ij} - m_j}{s_j} \tag{13}$$

$$m_j = \frac{1}{N}\sum_{i=1}^{N} I_{ij} \tag{14}$$

$$s_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(I_{ij} - m_j)^2} \tag{15}$$

D. Evaluation Criteria

For evaluating the IDS outputs (in the test phase), an IDS Evaluator component is added to IDCPSO. This component, by comparing the output of IDS and expected output of the system (which is determined for a test data set in a separate file), calculates the following evaluation metrics:

1) DR: true detection rate (only separating normal traffic from attacks);

2) FPR: false positive detection rate (mis-detecting attacks);

3) FNR: false negative detection rate (failing to detect attacks when they are occurred).

False negatives are dangerous problems, and are far more serious than the problem of false positives.

$$DR = \frac{the\ \ number\ \ of\ \ attacks\ \det ected}{the\ \ number\ \ of\ \ attacks} \tag{16}$$

$$FPR = \frac{the\ \ number\ \ of\ \ false\ \ positive}{false\ \ positive + true\ \ positive} \tag{17}$$

$$FNR = \frac{the\ \ number\ \ of\ \ false\ \ negative}{false\ \ negative + true\ \ negative} \tag{18}$$

E. Parameters Setup

In order to achieve the best performance of IDCPSO, we implement the experiments with different kinds of parameters setup. From the Fig.4, 5, we find that the convergence process of fitness function with c1=c2=2 is better than that of other value (c1=c2=4), and we can get the best value of fitness function when we set c1=c2 rather than c1≠c2. Determination of particle population size is still very important, if the value is set too small, we do not get the optimal value; conversely, if the value is set too large, the iteration time will be too long. Fig. 6 describes the relation between the population size and the average fitness value. Fig.7 shows the convergence process with different maximum velocity. We can obtain the best convergence when we set the maximum velocity equal to 10(vmax=10). Fig.8 explains the settings on the inertia weight value problem. The most reasonable setting of inertia weight value is w=1.0.

Therefore, the final parameters setup is showed in table IV.

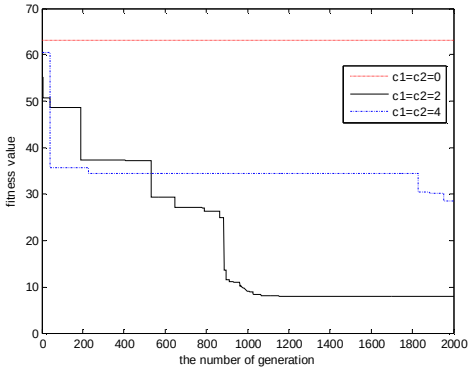

Figure 4. The convergence process when $c_1 = c_2$

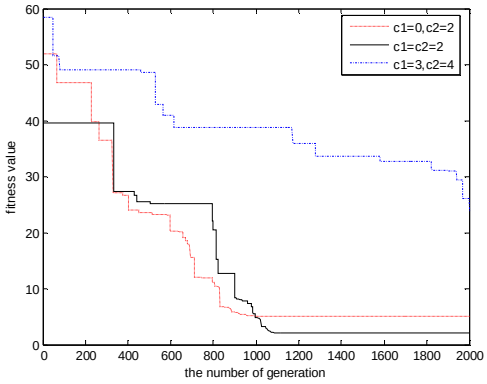

Figure 5.The convergence process using different acceleration coefficients value

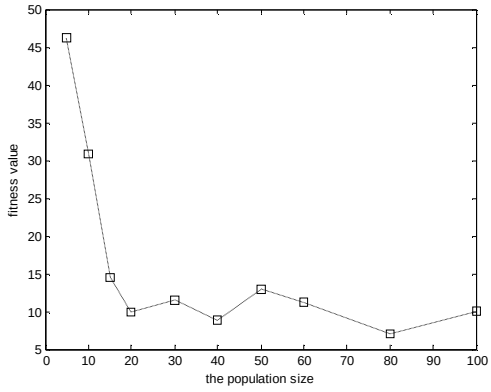

Figure 6. The relation between the population size and the average fitness value

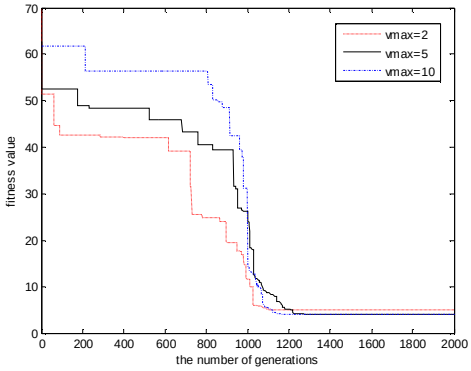

Figure 7. The convergence process with different vmax

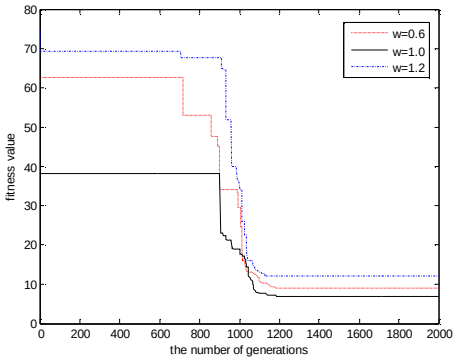

Figure 8. The convergence process with different inertia weight values

TABLE IV.
THE IDCPSO ALGORITHM PARAMITERS SET

Parameter	Value
$c_1 = c_2$	2
w	1
$v_{\max}$	10
Particles	20
Iterations	1500

F. Performance Comparison

To evaluate and compare the performance of the proposed algorithm, it is compared with Genetic Algorithm (GA), because GA is also population-based evolutionary computation algorithm which is widely used in many areas [36, 37, 38]. The comparison result about the convergence process of fitness function is given in Fig.9. It is obvious that the convergence speed of IDCPSO is faster and the number of iterations is less than that of GA. The detection performance is shown in Table V. According to the results, IDCPSO has higher DR, lower FPR and FNR than that of GA. We can obtain a better value about fitness function in much less generations, therefore, detection results is better than that of GA. In fig.10, 11, as the number of particles decreases, both the detection results and the convergence value decrease dramatically in GA; especially it is obvious when the number of particles is equal to 5. As for IDCPSO, we can find that not only the detection rate but also the convergence result are unchangeable. Therefore, the performance of GA is more sensitive to the number of particles than that of IDCPSO. Table VI shows the comparison of detection results with different particles. Table VII describes the initial class centers and the final class center after using PSO optimization of the first particle.

In addition, the performance of the IDCPSO algorithm also depends on the value of K; we varied K's value from 2 to 10.Fig.12 shows the ROC curves for 5 different K values. For this particular data set, K=8 is a better choice than other values in that the attack detection rate nearly reaches 98% and the false positive rate remains as low as 2%. The larger the K's value, the longer the algorithm's computing time. Therefore, we

can find compromise between the detection rate and the computing time when K equal to 8.

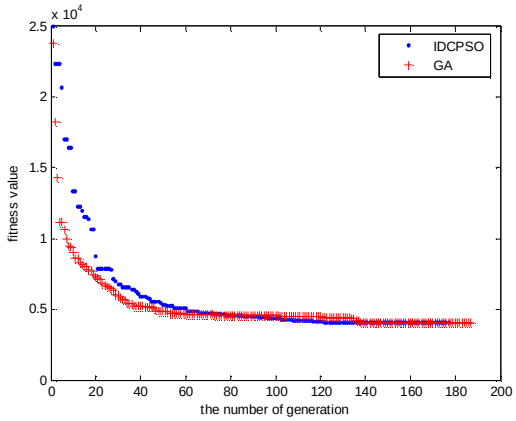

Figure 9. The comparison of fitness function results

TABLE V.
THE COMPARISON OF PERFORMANCE OF IDCPSO AND GA

	IDCPSO	GA
DR	0.967	0.933
FPR	0.023	0.171
FNR	0.033	0.066
Fitness function	4051.4	4239.2
The number of generations	176	190

TABLE VI.
THE COMPARISON OF DETECTION RESULTS WITH DIFFERENT PARTICLES

		DR	FPR	FNR
GA	particles =10	0.924	0.040	0.080
	particles =5	0.048	0.988	0.960
IDCPSO	particles =10	0.972	0.022	0.020
	particles =5	0.972	0.022	0.020

TABLE VII.
THE INITIAL CLASS CENTER AND FINAL CLASS CENTER OF THE FIRST PARTICLE

	The initial class center		The final class center	
	Normal	Attack	Normal	Attack
1	0.017607	-0.018325	-0.5400	1.5776
2	-0.098705	0.10273	0.1012	-0.2957
3	0.009729	-0.010126	-0.5771	1.6861
4	0.010499	-0.010928	-0.5769	1.6855
5	0.034638	-0.036052	-0.4478	1.3082
6	0.12934	-0.13462	-0.1104	0.3224
7	0.004035	-0.004200	-0.5290	1.5454

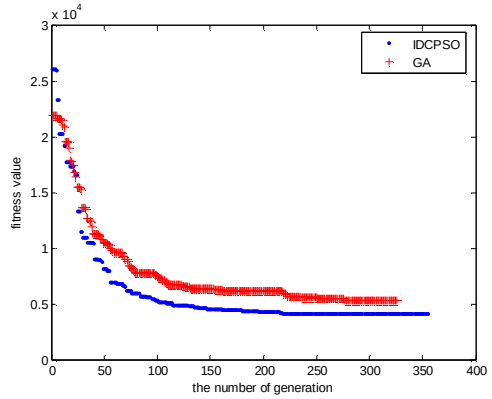

Figure 10. The convergence process of fitness function with particles=10

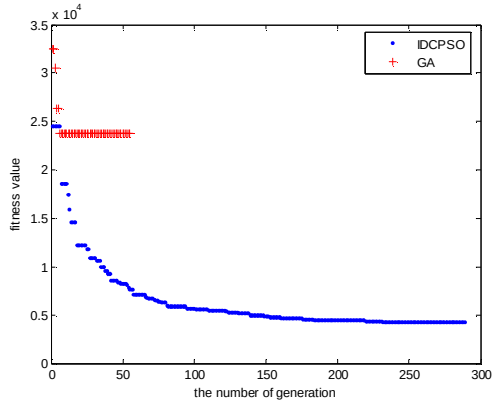

Figure 11. The convergence process with particles=5

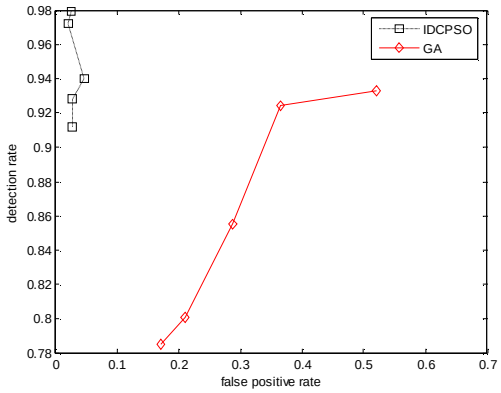

Figure 12. the ROC curve of IDCPSO and GA with different K

V. CONCLUSIONS

Prevention of security breaches completely using the existing security technologies is unrealistic. As a result, intrusion detection is an important component in network security. In this paper, IDCPSO has been proposed which combines the clustering with PSO algorithm. The simulation experiments compared with GA, indicate the following aspects:(1)we can obtain the global optimal value by using the IDCPSO algorithm;(2) detection rate is higher and FPR, FNR are lower than that of GA;(3)the speed of the convergence process is also higher than that of GA.

Our future work will focus on how to improve the detection rate on predicting attacks, especially the attacks of U2R and R2L. And due to the parameters' impact on the algorithm, we will consider introducing the other intelligent algorithms for the optimal parameters' selection.

ACKNOWLEDGEMENTS

The work in this paper is supported by the Natural Science Foundation of Chongqing, China (Grant No. 2008BB2182 and 2008BB0173), the Innovation Ability Training Foundation of Chongqing University, China (Grant No. CDCX021), and the National Natural Science Foundation of China (Grant No. 61070246).

REFERENCES

[1] Yuebin Bai, Hidetsune Kobayashi. Intrusion detection systems:technology and development.Proceedings of the17 th International Conference on Advanced Information Networking and Applications (AINA'03).

[2] L. Portnoy,E. Eskin,S. Stolfo. Intrusion detection with unlabeled data using clustering. In:Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001) Philadelphia. 2001:5-8.

[3] Nam Hun Park, Sang Hyun Oh, Won Suk Lee. Anomaly intrusion detection by clustering transactional audit streams in a host computer. Information Sciences 180 (2010) 2375–2389.

[4]Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai,Citra Dwi Perkasa. A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications 38 (2011) 306–313.

[5] Haidong Yang, Chunsheng Li , Jue Hu. RFID intrusion detection with possibilistic fuzzy c-Means clustering.Journal of Computational Information Systems, v 6, n 8, p 2623-2632, August 2010.

[6] Liang Hu, Nurbol, Xiaobo Liu, Kuo Zhao. A time stamped clustering method for intrusion detection. Journal of Information and Computational Science, v 7, n 2, p 399-406, February 2010.

[7]Panda, Mrutyunjaya,Patra, Manas Ranjan. A hybrid clustering approach for network intrusion detection using cobweb and FFT. Journal of Intelligent Systems, v 18, n 3, p 229-245, 2009.

[8] Pavel Laskov, Patrick D¨ussel, Christin Sch¨afer and Konrad Rieck. Learning intrusion detection:supervised or unsupervised?. 12489 Berlin, Germany.

[9]Y. G. Liu, K. F. Chen, X. F. Liao, Wei Zhang. A genetic clustering method for intrusion detection. Pattern Recognition. 37 (2004):927–942.

[10]Y. H. Liao, V. R. Vemuri. Use of K-nearest neighbor classifier for intrusion detection. Computers Security 2002;21:439–448.

[11]C. F. Tsai, C. Y. Lin. A triangle area based nearest neighbors approach to intrusion detection. Pattern Recognition, In Press, Corrected Proof, Available online 3 June 2009.

[12] W. H. Chen, S. H. Hsu, H. P. Shen. Application of SVM and ANN for intrusion detection. Computers & Operations Research, Volume 32, Issue 10, October 2005: 2617-2634.

[13] WE1 Yu-xin, WU Mu-qing. KFDA and clustering based Detection. The journal of china universities of posts and telecommunications. Volume 15, Issue 1, March 2008, pages: 123-128.

[14] Yang Yi, Jiansheng Wu, Wei Xu. Incremental SVM based on reserved set for network intrusion detection. Expert Systems with Applications, 38 (2011):7698-7707.

[15] Yao, Yu, Yang, Wei; Gao, Fu-Xiang; Yu, Ge. Anomaly intrusion detection approach using hybrid MLP/CNN neural network. In proceedings - ISDA 2006: Sixth International Conference on Intelligent Systems Design and Applications, 2006, v2, pages:1095-1102.

[16] Xiaobai Li. A scalable decision tree system and its application in pattern recognition and intrusion detection. Decision Support Systems 41 (2005):112–130.

[17] M. Amini, R. Jalili, H. R. Shahriari. RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks. Computers & Security, Volume 25, Issue 6, September 2006:459-468.

[18] Günes Kayacik, H. Nur Zincir-Heywood, A.; Heywood, Malcolm I. On the Capability of an SOM based Intrusion Detection System. Proceedings of the International Joint Conference on Neural Networks, v 3, pages:1808-1813, 2003.

[19] H.G. Kayacik, A.N. Zincir-Heywood, M.I. Heywood, A hierarchical SOM-based intrusion detection system, Engineering Applications of Artificial Intelligence 20 (4) (2007) 439–451.

[20] X. B. Tan, H. S. Xi. Hidden semi-Markov model for anomaly detection. Applied Mathematics and Computation, Volume 205, Issue 2, 15 November 2008:562-567.

[21] S. H. Oh, W. S. Lee. An anomaly intrusion detection method by clustering normal user behavior. Computers & Security. 2003.22(7): 596-612.

[22] K. Leung, et al. Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters. Proceedings of the Twenty-eighth Australasian conference on Computer Science, 2005:333-342.

[23] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, The VLDB Journal 16 (2007): 507–521.

[24] X. Cheng, P. C. Yong, L. S. Meng. Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. Pattern Recognition Letters. Volume 29 , Issue 7 (May 2008) :918-924.

[25] Hongying Zheng, Meiju Hou, Yu Wang. Application of Particle Swarm Optimization to Clustering for Intrusion Detection. The proceeding of 3rd international symposium on parallel architectures, algorithms and programming. Dalian, China,18-20 december 2010,Pages:221-228.

[26]D. E. Brown. C. L. Huntley. A Practical Application of Simulated Annealing to Clustering. Pattern Recognition,1992, 25(4):401-412.

[27]R. Eberhart, J. Kennedy. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micromachine and human science. Nagoya. 1995:39-43.

[28]J. Kennedy, R. Eberhart. Particle swarm optimization. In: Proceedings IEEE international conference on neural networks. Perth. 1995.1942-1948.

[29]Y.S. Jiang, J.X. Wang, H. Z. Yang. Attribute Discretization for Decision System Based on Binary Particle Swarm Optimization. Control Engineering of China:2008 V0l.15, No.4:360-363.

[30]C. J. Liao, C. T. Tseng, P. Luarn. A discrete version of particle swarm optimization for flowshop scheduling problems. Computers & Operations Research, Volume 34, Issue 10, October 2007:3099-3111.

[31]M. Maitra, A. Chatterjee .A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Systems with Applications, Volume 34, Issue 2, February 2008:1341-1350.

[32]H. Pan, L. Wang, B. Liu. Particle swarm optimization for function optimization in noisy environment. Applied Mathematics and Computation, Volume 181, Issue 2, 15 October 2006: 908-919.

[33]T. K. Rasmussen, T. Krink. Improved Hidden Markov Model training for multiple sequence alignment by a particle swarm optimization—evolutionary algorithm hybrid. Biosystems, Volume 72, Issues 1-2, November 2003:5-17.

[34]P. Y. Yin, S. S. Yu, P. P. Wang, Y. T. Wang. A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems. Computer Standards & Interfaces 28 (2006): 441–450.

[35]Lincoln Labs, KDD-cup data set. http://kdd.ics.uci.edu/databases/kddcup99.html.

[36]Louis Gosselin, Maxime Tye-Gingras, François Mathieu-Potvin. Review of utilization of genetic algorithms in heat transfer problems. International Journal of Heat and Mass Transfer, Volume 52, Issues 9-10, April 2009, Pages 2169-2188.

[37] N.F. Wang, K. Tai. Target matching problems and an adaptive constraint strategy for multiobjective design optimization using genetic algorithms. Computers & Structures, Volume 88, Issues 19-20, October 2010, Pages 1064-1076.

[38]Xiao-Ping Zeng, Yong-Ming Li, Jian Qin. A dynamic chain-like agent genetic algorithm for global numerical optimization and feature selection. Neurocomputing, Volume 72, Issues 4-6, January 2009, Pages 1214-1228.

BIOGRAPHIES

Hongying Zheng is an associated professor in the College of Computer Science, Chongqing University in Chongqing, P.R. China. She was born in 1975. She received her B.S. degree in application of computer and M.S. degree in architecture of computer from chongqing university in 1999 and 2002 respectively, and Ph.D. degree was received in information security. Her research interests mainly include information security, neural network, data mining and etc.

Meiju Hou was born in Henan,China,in 1987.She received the B.S. degree in computer science and technology from Zhengzhou university, Henan, China, in 2009. At present, she is working on M.S. degree in computer science, Chongqing university, Chongqing,China. Her main research interests focus on the application of computational intelligence in intrusion detection systems.

Yu Wang was born in Chongqing,China,in 1987. He received his Batcher's Degree from the Chinese People's Liberation Army Artillery Institute, Hefei, China in 2004. He is currently a postgraduate in the College of Computer Science of Chong University, Chongqing, China. His research interests include database system architecture and information security.