Grey self-organizing map based intrusion detection*

WANG Chun-dong（王春东）[1,2], YU He-feng（虞鹤峰）[1,2], and WANG Huai-bin（王怀彬）[1,2]**

1. School of Computer Science and Technology, Tianjin University of Technology, Tianjin 300191, China

2. Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin 300191, China

(Received 13 October 2008)

Grey self-organizing map (GSOM) model is proposed and applied in the detection of intrusion. Through the improvement of the weight adjustment using the GRC (grey relational coefficient), the training results of SOM get better. In the detection of deny of service (DOS) attacks, this model can consider the relativity of the data set of DOS attacks. Finally, the experiments on the DOS data set confirm their validities and feasibilities over this GSOM model.

Document code: A **Article ID:** 1673-1905(2009)01-0064-5

DOI 10.1007/s11801-009-8142-7

Most of the research institutes proposed the model of intrusion detection. The introduction of neural network raised the detection precision of DOS and using high efficient neural network model has become the key to DOS attacks. The Kohonen's self-organizing map(SOM) is a typical artificial neural network model and algorithm that implements a nonlinear feature projection from the high-dimensional space of signal data into a low-dimensional array of neurons in an orderly fashion[1,2].The mapping tends to preserve the topological relationships of signal domains. SOM has the ability of clustering, self-organizing, self-learning, visualization, classification etc. and it is widely applied in the fields of pattern recognition, data mining, incipient diagnosis and intrusion detection[3-6]. But its weight adjustment is determined only by its learning rate and the difference between the input pattern and the winner neuron's weight. It seems that the SOM obviously ignores some correlation relationships during the learning, which actually exist between the input pattern and the weights of all the nodes that participate in competition. Grey relational coefficient (GRC), which characterizes and stresses the aforementioned correlation relationships, was explicitly introduced into the learning rule of the traditional SOM by Hu Yi-Chung[7].

So in this paper, an improved SOM method, GSOM, is used in intrusion detection. The experiments on the benchmark data set confirm validities and feasibilities over this GSOM based intrusion detection.

The SOM network consists of two layers. The first layer is the input layer, which is responsible for the input; the second layer is the output layer, which is a node matrix with the input nodes lying below, as shown in Fig.1.

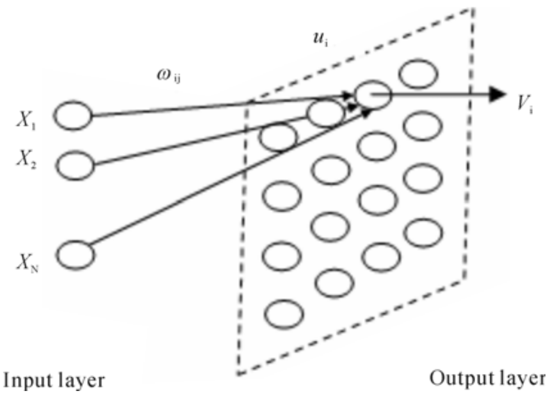

Fig.1 The model of SOM

The algorithm of SOM is recursive, as shown in Fig.2. First, every neuron corresponds to a N-dimension vector $W_i(k)=[w_{i1},w_{i2},...,w_{iN}]^T$. At every stage of training, sampling vector $X(k)=[x_1, x_2,...,x_N]^T$ is selected from the training set randomly, then the distance between X_k and all the weight vectors is calculated. c is the BMU(best-matching unit), and the minimum distance between c and X_k is

$$\|X_k - W_c\| = \min_{i=1}^{m}\|X_k - W_i\| \ . \tag{1}$$

Next, the weight vector of the neuron which is in neighborhood zone of the winner cell's topology is updated. The rule is as follows:

* This work has been supported by the Tianjin Key Subject Fund, and the Higher Education Science and Technology Development Fund of Tianjin Municipal Education Commission (No. 2006BA19).

** E-mail: michael 3769@163.com

$$W_i(k+1)=\begin{cases}W_i(k)+\alpha(k)[X_k-W_i(k)] & i\in N_c(k)\\ W_i(k) & i\notin N_c(k)\end{cases} . \quad (2)$$

In equation(2), N_c refers to neighborhood zone of the centre neuron W_c. In the process of learning, the initialization of $N_c(k)$ can be large, then contracts gradually, as follows:

$$N_c=\mathrm{INT}(N_c(0)(1-k/L)),\quad k=0,1,2,...L\ . \quad (3)$$

In equation(3), $N_c(0)$ means the initial neighborhood radius, L, the times of the iteration, INT(.), the integral function. $N_c(k_1)$, $N_c(k_2)$ and $N_c(k_3)$ stand for the topology neighborhood zone of the winner cell whose iterative times are k_1 , k_2 and k_3 ($k_1 < k_2 < k_3$).

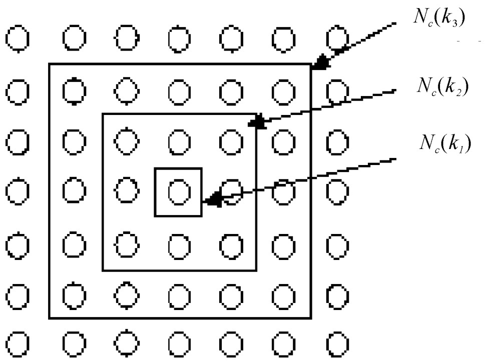

Fig.2 Topology adjacent domains on two- dimension network

Usually, the learning rate $\alpha(k)$($0< \alpha(k) <1$) is a constant, which is close to 1.0 at the beginning, then lessens gradually. For example, at can be 0.8(1-*k*/*L*). With the increase of the times of the iteration, $\alpha(k)$ tends to zero, which ensures the learning process to refrain from rash action.

The concrete steps of the learning algorithm of SOM are as follows [8-10]:

Step 1. Setting variables and parameters: Let $X(k)=[x_1, x_2,...,x_N]^{\mathrm{T}}$ be the input vector, or train $W_i(k)=[w_{i1},w_{i2},...,w_{iN}]^{\mathrm{T}}$ be the weight vector, i=1,2, …,M, and the total times of iteration be L.

Step 2. Initialization: Initialize the weight vector W_i with a small random number in a certain interval. Let the neighborhood radius be $N_c(0)$; the learning rate be $\alpha(k)$; and then normalize weight vector $W_i(0)$ and all the input vector X.

$$X'=\frac{X}{\|X\|}\ , \quad (4)$$

$$W_i'(0)=\frac{W_i(0)}{\|W_i\|}\ . \quad (5)$$

In the above formulas, $\|W_i(0)\|=\sum_{j=1}^{N}[w_{ij}(0)]^2$ and $\|X\|=\sum_{j=1}^{N}(x_i)^2$ are Euclidean norm of the weight vector and input vector.

Step 3. Data sampling: Select training samples X′ from the input space.

Step 4. Approximate matching: According to the standard of the minimum Euclidean distance:

$$\|X'-W_c'\|=\min_i\|X'-W_i'\|\quad i=1,2,...,M\ , \quad (6)$$

Select winner cell c, and implement the competitive process of neurons.

Step 5. Updating (1): Update the weight vectors of the cordial neuron, which is in the topology neighborhood zone of the winner cell $N_c(k)$ under the following rules:

$$W_i'(k+1)=W_i'(k)+\alpha(k)h_{i,j}(k)[X-W_i'(k)]\ , \quad (7)$$

where $h_{i,j}(k)$ is a neighborhood updating function, called Maxican-Hat Function.

Step 6. Updating (2): Update the learning rate $\alpha(k)$ and the topology neighborhood zone, and then normalize the weights after learning.

$$\alpha(k)=\alpha(0)(1-\frac{k}{L})\ , \quad (8)$$

$$N_c(n)=\mathrm{INT}\left[N_c(0)\left(1-\frac{k}{L}\right)\right]\ , \quad (9)$$

$$W_i'(k+1)=\frac{W_i(k+1)}{\|W_i(k+1)\|}\ . \quad (10)$$

Step 7. Judging: Judge whether the times of the iteration k exceed L or not, if k<=L then turn to stage 3, otherwise end the process of iteration.

During the training, output neurons are sorted through adjusting their weight vector, in order to be close to the probability density. Though produced randomly at the beginning, the weight vector of the output neurons gets closer and closer to the distribution of the input data after long time running, through updating the weight vector continuously.

Grey system theory was put forward by Prof. Deng Julong in 1982[11]. During the development for nearly two decades, grey system theory has built up a new study structure.

Grey correlation analysis is used to quantitatively compare or depict the relative change between systems or every factor of the system, while the approximate extent of the size, direction and speed of the change is used to measure the relevance between them[12]. In this paper, grey correlation analysis refers to the research method of the correlation between a given reference pattern and another comparative pattern, both of which must be standardized[7]. Let input pattern X be the reference pattern, weight vector $W_i(i=1,2,\ldots M)$ be the comparative pattern, so as to analyze the grey correlation between X and W_i. And let the grey relational coefficient between the component of the input pattern $x_j(j=1,\ldots,N)$ and weight value w_{ij} be ξ_{ij}[7,13,14,15]:

$$\xi_{ij} = \frac{\Delta_{\min} + \lambda\Delta_{\max}}{\Delta_{ij} + \lambda\Delta_{\max}} . \tag{11}$$

In equation (11),$\lambda(0 \leqslant \lambda \leqslant 1)$ is the discrimination coefficient, which usually takes 0.5. Furthermore, $\Delta_{\min}$, $\Delta_{\max}$ and Δ_{ij} can be defined as follows:

$$\Delta_{\min} = \min_{i=1,\cdots,M} \min_{j=1,\cdots,N} \left| x_j - w_{ij} \right| \quad , \tag{12}$$

$$\Delta_{\max} = \max_{i=0,1,\cdots,M} \max_{i=0,1,\cdots,N} \left| x_j - w_{ij} \right| \quad , \tag{13}$$

$$\Delta_{ij} = \left| x_j - w_{ij} \right| \cdot \tag{14}$$

In equation (11), we can see $0 < \xi_{ij} \leqslant 1$, and if ξ_{ij} is close to 1, then Δ_{ij} will be close to $\Delta_{\min}$. That is to say, the closer the difference between the component of input pattern x_j and the weight value w_{ij} is, the closer the difference between every component of the input pattern X and the weight vector W_i will be. So, the grey relational coefficient ξ_{ij} reflects the extent of similar between x_j and w_{ij} .

Considering this character of ξ_{ij}, ξ_{ij} can be used in the learning rule of traditional model of SOM, thereby let the learning rule be more reasonable and effective. So, the learning rule of grey SOM is as follows:

$$\Delta w_{ij} = \alpha(\xi_{ij})^t (x_j - w_{ij}) \quad , \tag{15}$$

where t is a pre-specified positive real number, a is the learning rate.

The algorithm of grey SOM is as follows:

Input data: A given set of pixels.

Output data: The center of each cluster.

Step 1: Initialize connection weights and parameters.

1) Initialize weights corresponding to each output node with random small values;

2) Initialize $\alpha(0)$ and the number of neighbor nodes $N_i(0)$ of node i: $\alpha(0)$ should approach 1.0, and $N_i(0)$ should cover all output nodes, $1 \leqslant i \leqslant M$

3) Set $k = 1$, where t is an iteration counter.

Step 2: Present input training data $X(k)$.

Step 3: Determine the winning node c.

If $\| X(k) - W_c \| = \min\limits_{i=1}^{M} \| X(k) - W_i \|$, the node c will be the winner.

Step 4: Adjust the winning nodes c and its neighbor nodes.

1) The neighbor nodes around the winning node c can be determined by $N_i(k)$;

2) The learning rule based on $\xi_{ij}(k)$ can be given as equation (11).

$$w_{ij}(k+1) = w_{ij}(k) + \alpha(k)[\xi_{ij}(k)]^t [x_j(k) - w_{ij}(k)]$$
$$i \in N_i(k), 1 \le j \le N \ ,$$

where t is a pre-specified positive real number, and $\xi_{ij}(k)$ is the GRC between $x_j(k)$ and $w_{ij}(k)$. If each training data is presented to the network, then go to Step 5; otherwise go to Step 2.

Step 5: Shrink the learning rate $\alpha(k)$ and the neighborhood size $N_i(k)$.

Step 6: Convergence test.

If the winning node of each input data does not change, then stop the process. Otherwise, set k+1 to k and go to Step 2.

Like SOM, GSOM also includes two phases: sorting phase and convergence phase. Sorting phase is the updating phase of weight vector topology, the initial learning rate is 0.9, finally becomes 0.05, and the neighborhood zone changes from 8 down to 1. When it comes to the convergence phase, the statistic behavior of the input space is needed, so the feature mapping can be updated further. The initial learning rate is 0.05, finally becomes 0.01, and the neighborhood zone is 0 from the beginning to the end. In order to get the convergence result, the learning time of sorting phase should be longer than 2000; but when it comes to convergence phase, in order to save the expenses of algorithm, the learning time should be shorter than 1000.

In order to train GSOM and test its ability in intrusion detection to identify the DOS attacks, we choose some dataset from the DARPA'98 training data and testing data, shown as the Tab.1 below:

Tab.1 Data of different DOS attacks chosen from DARPA'98

Data	Types of DOS attack					
	Normal	Neptune	Smurf	Pod	Land	Teardrop
Training data	4863	10720	280791	264	21	980
Testing data	60593	58001	164091	87	9	12

The total number of training data samples is 297639, and testing data samples is 282793. In order to deal with the neural network, data preprocessing and standardization are necessary. But the data sampling is too larger, so how to choose the network topology of GSOM becomes the key to the problem.

Because of the large number of samples, the choice of

SOM topology becomes very important, which will cause the deviation of clustering and classification results. So, the principal component analysis (PCA) is used to define the topology of SOM in Ref.[16].

First of all, the number of SOM neurons should be determined. Define the whole training data as *dlen*, number of SOM neurons as *munits*. So, *munits* =1.25 × *dlen* ^ 0.54321.

This makes enough neurons project on the large primitive data. When the number of neurons is determined, PCA is used to determine the length and wide proportion of SOM.

The primitive dimension of vector is supposed to be n, namely $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)^T$. Another n new characters, $y_1, y_2 \cdots y_n$ should be constructed, and let them satisfy the conditions as shown below:

1) Every new character is the linear combination of primitive characters, namely

$$y_i = u_i^T X \quad u_i = (u_{i1}, u_{i2}, \cdots, u_{in})^T, i=1,2,\cdots n \ .$$

2) Every new character is irrelevant, namely the correlation coefficient is 0.

$$r(y_i, y_j) = 0 \quad i, j = 1,2,\cdots n, i \neq j \ .$$

3) u_i makes the covariance of y_i maximize, $i = 1,2,\cdots,n$.

The new characters $y_1, y_2 \cdots y_n$ are the principal components of node $1,2,\cdots,n$. In order to calculate every principal component, we only need to calculate every u_i, which can be proved as the feature vector of the sample covariance matrix $\boldsymbol{S}_x$.

$$\boldsymbol{S}_x = \boldsymbol{E}[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T] \ . \tag{16}$$

Let $\boldsymbol{U}=(u_1, u_2, \cdots u_n)^T$, content with orthogonal normalization, $\boldsymbol{UU}^T = \boldsymbol{I}$, then

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_3 \end{bmatrix} = \boldsymbol{UX}$$

$\boldsymbol{S}_x\boldsymbol{U}^T = \boldsymbol{U}^T\boldsymbol{\Lambda}$, in which $\Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix}$ is the characteristic value matrix. λ_i is mapping to the characteristic vector u_i.

After the change, if $\boldsymbol{X}$ is mapping to the covariance matrix $\boldsymbol{S}_x$, then the mapping covariance matrix of $\boldsymbol{Y}$ is:

$$\boldsymbol{S}_y = \boldsymbol{E}[\boldsymbol{YY}^T] = \boldsymbol{US}_X\boldsymbol{U}^T = \boldsymbol{UU}^T \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix} \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix} \tag{17}$$

That is to say, the covariance between every two new characters $y_1, y_2 \cdots y_n$ is 0, namely they are irrelevant. The covariance reflects the incorporative information in some sense. Sort the characteristic value, let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, when the information of the former m new character is enough, namely $\sum_{i=1}^{m} \lambda_i \Big/ \sum_{i=1}^{n} \lambda_i \geq 98\%$, the other new character can be discarded.

So, $\psi = \lambda_1 / \lambda_2$ is defined as the length and width of proportion of SOM. The topology of SOM can be determined through larger number of training data and characteristic value.

In the experiment, the training data is input in the model of GSOM. From the results of experiment 1) and experiment 3), Tangent is chosen to be the function format of grey relational coefficient. The choice of experiment parameters is as follows:

According to the analysis above, the choice of SOM topology should be 81 × 6;

Sorting phase: the largest time of learning is 50, the initial learning rate is 0.9, the last learning rate is 0.05, the initial value of neighborhood zone is 8 and the last is 1;

Convergence phase: the largest time of learning is 40, the initial learning rate is 0.05, the last learning rate is 0.01, the initial value of neighborhood zone is 0 and the last also is 0;

Tangent parameters are: $\beta = 1, \theta = \pi$;

After training, the testing data set is used to test GSOM, experiment results can be seen in Tab.3 and Tab.4. Because of the mapping mechanism of SOM, the label of neurons on the class border can be not determined. So, let the label be 0, labeled as "unknown" in Tables.

Something should be especially pointed out: the dimension of some data character is greater, but these dimensions reflect the main data character of this kind of attack. If normalizing these dimensions, the integrity of the primitive character will be broken.

$$M_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} \qquad i = 1,2,\cdots N \quad j = 1,2,\cdots 41 \ , \tag{18}$$

$$C_j = \frac{1}{N-1}\left(\sum_{i=1}^{N}\left(x_{ij} - M_j\right)^2\right)^{\frac{1}{2}} \quad i = 1,2,\cdots,N \quad j = 1,2,\cdots,41 \quad , \quad (19)$$

$$X_{ij} = \frac{x_{ij} - M_j}{C_j} \quad i = 1,2,\cdots,N \quad j = 1,2,\cdots,41 \ , \quad (20)$$

where M_j is the average value of column *j*, C_j is the covariance of column *j*, and X_{ij} is the new value after normalization. The experiment results of Tab. 3 are the results after normalization, shown below.

In Tab.2, the normalization breaks the primitive character; which makes the classification of DOS attacks and Normal data bad. Tab.3 and Tab.4 are the classification results of SOM and GSOM without normalization.

Tab.2 GSOM based detection results of DOS attack (normalization)

	Normal	Neptune	Smurf	Pod	Teardrop	Unknown	Total	Correct (%)
Normal	20856	4	0	0	70	40012	60942	34.22
Neptune	2064	16388	0	0	0	38980	57432	28.53
Smurf	31818	0	130299	0	0	320	162437	80.21
Pod	5	0	0	0	78	0	83	0
Teardrop	0	0	0	0	0	11	11	0

In Tab.3, SOM based detection rate of Normal is 95.77%, the detection rate of Neptune, Smurf, Pod and teardrop are respectively 97.90%, 99.98%, 100% and 75%. The average detection rate is 93.73%.

Tab.3 SOM based detection results of DOS attack (not normalization)

	Normal	Neptune	Smurf	Pod	Teardrop	Unknown	Total	Correct (%)
Normal	58302	77	667	250	858	723	60877	95.77
Neptune	58	55999	0	0	983	158	57198	97.91
Smurf	0	0	163989	0	0	18	164007	99.98
Pod	0	0	0	85	0	0	85	100
Teardrop	4	0	0	0	12	0	16	75

In Tab.4, GSOM based detection rate of Normal is 97. 65%, the detection rate of Neptune, Smurf, Pod and teardrop are respectively 99.67%, 99.99%, 91.66% and 100%. The average detection rate is 97.794%. So, the detection ability of GSOM is better.

Tab.4 GSOM based detection results of DOS attack (not normalization)

	Normal	Neptune	Smurf	Pod	Teardrop	Unknown	Total	Correct (%)
Normal	60012	104	159	48	350	778	61451	97.65
Neptune	0	58212	0	0	192	0	58404	99.67
Smurf	0	0	163989	0	0	4	163993	99.99
Pod	5	0	3	88	0	0	96	91.66
Teardrop	0	0	0	0	15	0	15	100

In contrast to SOM, the detection rate of GSOM declines by 8.34%. But the average detection rate of the system rises 4.064%. From the analysis above, it can be concluded GSOM has better detection ability than SOM on the whole.

This paper introduces IDS, analyzes the data character of DOS attack and proposes GSOM model using grey relational coefficients and its correlative functions. The purpose is to make the input pattern and the weight value of the competitive neurons as a whole, to express the inner relationship between them through general grey relational function, and melt the inner relationship into the learning rule of SOM, thus the capability of the algorithm is improved further. In contrast to SOM based IDS, GSOM based IDS largely raises the detection precision of DOS attacks.

References

[1] T.Kohonen, Self-organization and Associatiue Memory, Springer-Verlag Berlin Heidelberg, New York, 1989.

[2] T.Kohonen, Proceedings of the IEEE, **78** (1990), 1464.

[3] J.Iivarinen, T.Kohonen, J.Kangas, S.Kaski, in proceedings of Conference on Artificial Intelligence Research in Finland, Helsinki, (1994), 122.

[4] D.Merkl, A.rauber, in Proc. Workshop on Self-Maps (WSOM97), Helsinki, Finland,1997.

[5] J.Vesanto, E.Alhoniemi, IEEE Transactions on Neural Networks, **11** (2000), 586.

[6] D.Alahakoon, S.K.Halgamuge, B.Srinivasan, Dynamic IEEE Transaction on Neural Networks, **11** (2000), 601.

[7] Hu Yi-Chung, Chen Ruey-Shun, Hsu Yen-Tseng, Neurocomputing, **48** (2002), 863.

[8] Sahin Albayrak, Achim Muller, Christian Scheel, Dragan Milosevic, In Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05), 2005

[9] H. Giinev Kayacik, A. Nur Zincir-Heywood, Malcolm I. Heywood, In IEEE International Joint Conference on Neural Network, (2003), 1808.

[10] Jianhong Gao, Lixin Xu, Yaping Dai, In Proceedings of the 5th World Congress on intelligent Control and Automation, (2004), 4367.

[11] J.L.Deng, Systems Control Lett, **1** (1982), 288.

[12] Liu Si-feng, Guo Tian-bang, Dang Yao-guo, Grey System Theory and Application, Science Press,1999.

[13] K.Q.Shi, G.W.Wu, Y.P.Hwang, Theory of Grey Information Relation, Chuan Hwa, Taiwan,1994.

[14] C.C.Cheng, Y.T.Hsu, C.C.Wu, IEICE Transactions on Fundamentals Electronics, Commun. Comput.Sci.1998,E81-A (11), 2433.

[15] Tan Zhi-guo,Chen Song-chan, Journal of Nanjing University (Natural Sciences), **38** (2002), 63.

[16] Somtoolbox, http://www.cis.hut.fi/projects/somtoolbox.