

Network intrusion clustering based on fuzzy C-Means and modified Kohonen neural network

Hongwei Ye^{1*}, Lianjiao Zhang¹, Xiaozhang Liu²

¹School of Electron and Information Engineering, Heyuan Polytechnic, Dong Huan Road, Heyuan, China

²School of Computer Science, Dongguan University of Technology, No.1, Daxue Rd Songshan Lake, Dongguan, China

Received 6 July 2014, www.cmmt.lv

Abstract

Kohonen neural network recognizes and clarifies substantive network data, but with a long running time and a slow convergence process. To solve this problem, a network intrusion clustering method is presented in this paper. Specifically, the training data is pretreated using Fuzzy C-Means (FCM). Then some selected data will be trained with using Kohonen neural network. Meanwhile, to speed up the convergence process of Kohonen neural network and to form a better optimized network topology, a neighbourhood function is established for the competing neuron. Each neuron has neighbourhood topology collections. The data simulation results demonstrate the efficiency and effectiveness of the proposed algorithm.

Keywords: Kohonen neural network; neuron, FCM, network intrusion clustering

1 Introduction

Network intrusion is an unauthorized access to the completeness, privacy and availability of a computer or network. Intrusion detection is an intrusion detection system that attempts to discover unauthorized malicious activity by analysing information collected by the computer or network. Kohonen neural network (KNN) is applied to the detection of network intrusion. The improvement of KNN can be achieved by a set suitable threshold to the competition layer in the neural network to avoid the occurrence of necrotic neuron [1]. Also fuzzy bias degree is introduced in some documents, which aims to avoid dead neurons [2]. To speed up the convergence process of KNN, energy function is introduced [3]. An initial value is set by a simple splitting algorithm to improve KNN. The more data it has, the longer time KNN training will take. To speed up the process, each kind of training is FCM clustered. Data is chosen according to the degree of membership for KNN training. Meanwhile, KNN will be improved in two aspects. One is neighbourhood function established among neurons, and the other is topology identification of neural neighbours, in which the training data, after being correctly clarified, reach suitable topology and then the iteration breaks.

2 FCM

FCM is a cluster algorithm based on objection function. It is developed from Hard C-Means (HCM). It is a algorithm that determines the likelihood of a data by using the membership degree. If X is a vector of a limited data set in eigenspace R^N . According this algorithm, X is divided

into several fuzzy cluster, each centre vector forms a set $V = \{v_1, v_2, \dots, v_c\}$, a matrix of $N \times C$ dimension $U = (u_{ij})$, $u_{ij} \in [0, 1]$, a membership matrix of each sample, and here $i = 1, 2, \dots, N$; $j = 1, 2, \dots, C$ makes the fuzzy objective function minimum. Fuzzy clustering objective function is presented in Equation (1) and Equation (2).

$$J = \sum_{i=1}^N \sum_{j=1}^C u_{ij} \|x_i - v_j\|^2, \quad (1)$$

$$u_{ij} = \begin{cases} \left[\frac{\sum_{k=1}^c \|x_i - v_j\|^{\frac{2}{m-1}}}{\sum_{k=1}^c \|x_i - v_k\|^{\frac{2}{m-1}}} \right]^{-1} & \|x_i - v_k\| \neq 0 \\ 1 & \|x_i - v_k\| = 0 \&\&k = j \\ 0 & \|x_i - v_k\| = 0 \&\&k \neq j \end{cases} \quad (2)$$

$$v_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m}$$

Here, u_{ij} is the degree of membership for the x_i in cluster j , m is the fuzzy weight index, v_j is the centre vector for cluster j .

FCM algorithm schema:

Step 1: fix discriminant function C , and fuzzy weight index m .

Step 2: initial centre vector for cluster v .

*Corresponding author e-mail: aboyhw@163.com

- Step 3: compute the membership matrix u .
- Step 4: compute each cluster centre v .

Step 5: compute the fuzzy cluster objective value to determine if it meets the given termination conditions. If it meets, the iteration ends. If not, jump to step 3.

3 Modified Kohonen neural network

The model was first described as an artificial neural network by the Finnish professor Teuvo Kohonen in 1981, and is sometimes called a Kohonen map or network. It is sometimes called SOFM [5]. It automatically gets the centre of each cluster by competitive learning training coefficient, so it is widely used in fields like pattern identification and patten control. Kohonen believes that neurons in different part do their own jobs [6]. Different parts of the network will respond accordingly to certain input patterns. Kohonen network can learn both the distribution characters and the topology structure of input vector of training data [7, 8]. Typical structure of KNN is shown in Figure 1.

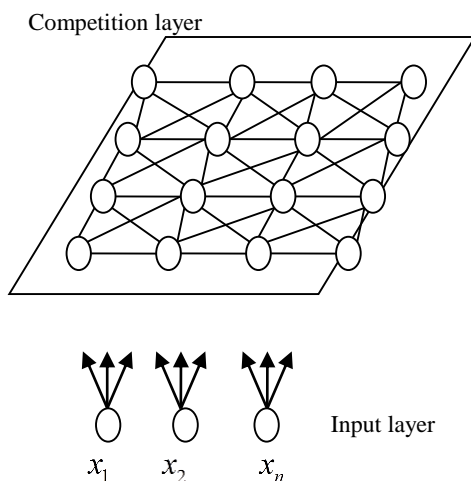


FIGURE 1 The structure of KNN

The training algorithm for the Kohonen network as follows:

(1) Initialise the weight vectors, $w_j(0)$, where j is an index denoting the neuron number and runs form $1, 2, \dots, K$, where K is the total number of neurons int the output grid. The number in () afterwards is used to denote the time-step. The weight vectors can be initialised by setting each component of each weight vector to a small random number.

(2) Draw a connection vector, $X = (x_1, x_2, \dots, x_m)^T$, from the training data without replacement.

(3) Find the winning neuron, j^* , This is the neuron whose weight vector is closed to X in a Euclidean sense, calculated as follows:

$$d_j = \|X - W_j\| = \sqrt{\sum_{i=1}^m (x_i(t) - w_{ij}(t))^2}$$

- (4) Adjust the weight vectors of all neurons, as follows:

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(t)(x_i(t) - w_{ij}(t))$$

where $\eta(t)$ is the learning rate at epoch number t , usually $\eta(t) = \frac{1}{t}$ or $\eta(t) = 0.2 \left(1 - \frac{t}{T}\right)$, T is the total number of training.

- (5) Repeat from step 2 until all training examples have been presented. This constitutes on epoch.

Repeat form step 2 until the desired number of epochs is reached.

According to the biology principal of neurons, neurons will be in a sorted order after self-organized training. Neurons of the same class will be next to each other, while those different will be in a distant [9-12], as Figure 2 shows. For example, a competition layer with 4 rows and 4 columns has 16 neurons. Among them, number 1 is the winning neuron, and then number 2 and number 5 neurons is more possible to win. But number 16 neuron is not likely to win. So, the first a probability matrix needs to be established for each neuron before network training.

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix},$$

where, b_{ij} is the transit probability between the i -th neuron and the j -th neuron. $i=1, \dots, n, j=1, \dots, n$. Here n is the total amount of the neurons in the competition layer. Then the neighbourhood function $G(t)$, follows a Gaussian probability distribution as defined by Equation (3), where $j(t)$ denotes the neuron in the neighbourhood of the winning neuron, and then adjust the weight vectors of all neurons as given in Equation (4).

$$G(t) = \exp\left[-\frac{(j(t) - j^*)^2}{2}\right], \tag{3}$$

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = g(t)\eta(t)(x_i(t) - w_{ij}(t)). \tag{4}$$

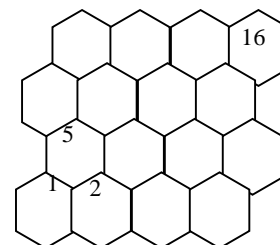


FIGURE 2 Topology of neurons in competition layer with 4 × 4

Identification of neighbour topology. Build a neighbour set U for each neuron. For example, the

neighbour set of number 1 neuron is {2,5}, and number 5 is {1,2,6,9,10}. In the process of KNN training, we set minimum learning times. If the training times exceed the set one, KNN begins to self-classification and identification with the aim to see if the neurons are correctly classified. For example, in a competition Kohonen neural network with 6 rows and 6 columns, there are 5 training kinds. The winning neuron in each kind forms a set, namely {6,12}, {7,13}, {16,21,22}, {25,31}, {30,36}. Take the intersection of each two sets. If all of them are disjoint, then it shows that KNN can classify correctly. At the same time, conduct neighbourhood topology identification.

4 Simulation results

4.1 GET DATA

Intrusion Detection has become a pertinent part of network security. An Intrusion Detection System can monitor the events happening in a system, and identify whether they are attacks or legitimate accessed. There are nine ‘basic features’ in Intrusion Detection System, such as duration of the connection; total bytes sent to destination host; total bytes sent to source host; service type ,such as FTP, HTTP, Telnet; number of wrong fragments. In our experiment, we get five attack categories as shown in Table1.

TABLE 1 Category attack data use by our experiment

Category	Description
1	HTTP
2	FTP
3	Mail
4	SQL
5	Remote shell

4.2 FILTER DATA

The attack dataset is $N_{4500 \times 39}$, we use the 4000 data as the training data, the last 500 as the test data. Each given 5 classified 4000×38 training data is FCM clustered. A membership matrix of each data will be formed. Arrange the matrixes in the order of membership degree. Choose the first 20 data as representatives. The algorithm of FCM clustered is described as Algorithm 1.

Algorithm 1: Filter data using FCM

Input: $N_{4500 \times 39}$

Output: TrainData_{100×38}

```

for i=1:category
{
DataIndex=find(T1==i);
data=inputn(DataIndex,:);
[center,U,obj_fcn] = fcm(data,1);
[su,index]=sort(U,'descend');
FilterData{i,:}=inputn(DataIndex(index(1:20)),:);
}
TrainData=[FilterData{1,1};FilterData{2,1};
FilterData{3,1};FilterData{4,1};FilterData{5,1}];
    
```

4.3 TRAINING KNN

We get a training data of 100×38. The dimension of the data is 38 from 5 different network invasion patterns. So there are 38 nodal points in the input layer. The nodal point in the competition lay represents the potential kinds of the input numbers. Usually the nodal point in the in the competition is far more than the actual data kinds. Here the author assumes the nodal point is 36 and the neuron topology arrangement is as shown in Figure 3.

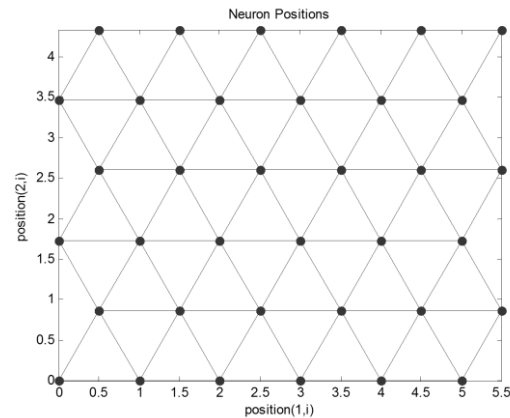


FIGURE 3 Topology arrangements of neurons in the competition layer

The data will be trained by the modified KNN (MKNN). The Process of simulation experiments as shown in Figure 4. The winning neurons’s topology arrangement is as shown in Figure 5.

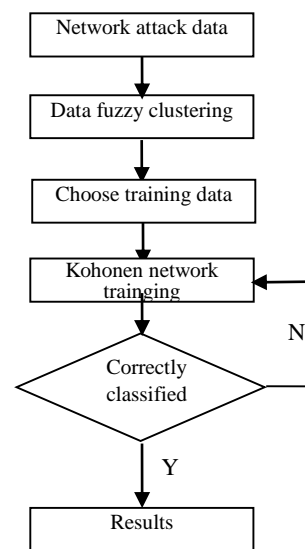


FIGURE 4 Process of simulation experiment

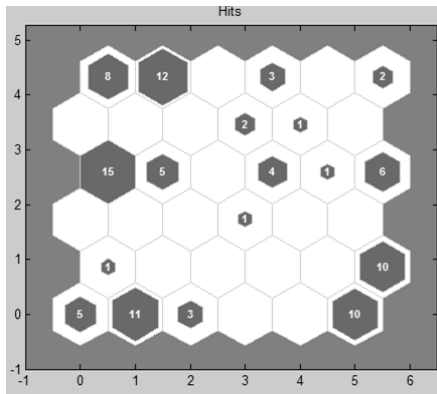


FIGURE 5 The winning neuron topology

The List of winning neurons as shown in Table 2. The computer adopted in the experiment is Intel Core i3-

2370M 2.4G with 6G RAM and a runtime environment of Matlab R2012b. It takes the traditional KNN 30 minutes and 42 seconds to deal with the training data, while only less 5 minutes using the method mentioned in this essay and the topology of the winning neurons are better than those of the traditional. The traditional KNN Cluster and the MKNN Cluster is shown in Figure 6 and Figure 7.

TABLE 2 List of winning neurons

kind	number
The first kind	6,12
The second kind	31,32
The third kind	19,20
The fourth kind	1,2,3,7
The fifth kind	16,22,23,24,28,29,34,36

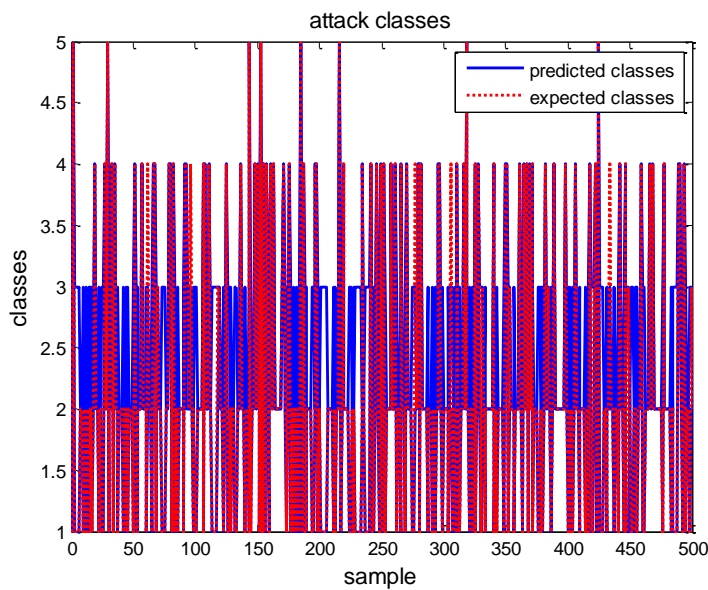


FIGURE 6 The traditional KNN Cluster

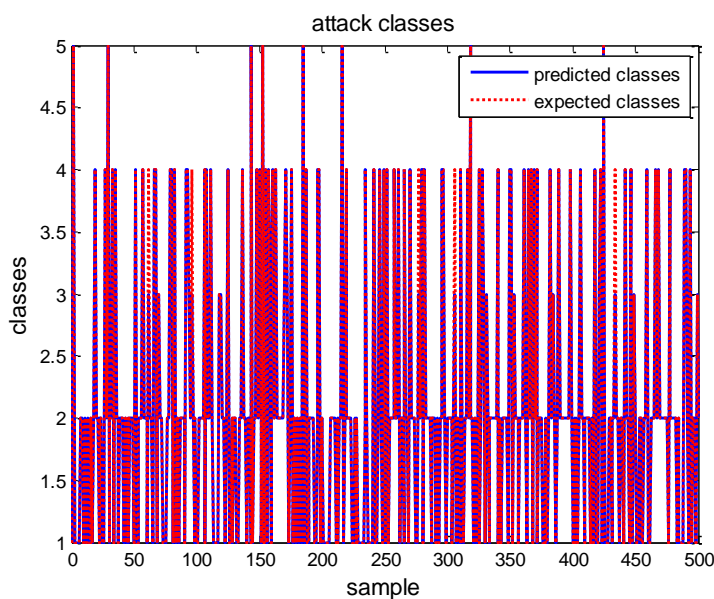


FIGURE 7 The MKNN Cluster


9 Conclusions

Based on the high clarification efficiency of KNN, this essay presents a method of FCM clustering in case of large amount of network invasion data. Typical data is chosen to be trained, which greatly cut down the training time of

network classification. Also the introduction of neural neighbourhood function and neural neighbourhood topology identification improves the neural topology distribution and classifies the data more accurately. The result of the simultaneous experiment shows the efficiency of the algorithm in network invasion clustering.

References

- [1] Fang H 2013 Track correlation algorithm based on modified Kohonen neural network *Journal of Computer Applications* **33** (5) 1476-80
- [2] Xu M-j 2009 Improved clustering algorithm based on fuzzy Kohonen clustering network *Computer Simulation* **26** (4) 228-32
- [3] Lei H 2005 Identification of porn images based on improved kohonen neural network and BP Network *Computer Engineering* **31**(10) 164-7
- [4] Liu Y 2011 The research of fuzzy clustering algorithm of data mining based on shuffled frog leaping algorithm *LanZhou:LanZhou University of Technology*
- [5] Yang C 2010 The research of kohonen neural network in telecom frauds forecast *ShangHai:EastChina Normal University*
- [6] Duan L-z, Zhu M, Wang L-m 2009 A web user access pattern mining algorithm based on the dual kohonen neural Network *Computer Engineering & Science* **31**(9) 95-8
- [7] De Almeida C W D, De Souza R M C R 2013 Fuzzy Kohonen clustering networks for interval data *Neurocomputing* 65-75
- [8] Bianchi D, Calogero R, Tirozzi B 2007 Kohonen neural networks and genetic classification *Mathematical and Computer Modelling* **45**(1) 34-60
- [9] Obimbo C, Zhou H, Wilson R 2011 Multiple SOFMs working cooperatively in a vote-based ranking system for network intrusion detection *Procedia Computer Science* **6** 219-24
- [10] Kayacik H G, Zincir-Heywood A N 2007 A hierarchical SOM-based intrusion detection system *Engineering Applications of Artificial Intelligence* **20** 439-51
- [11] Powers ST, He J 2008 A hybrid artificial immune system and Self Organising Map for network intrusion detection *Information Sciences* **178** 3024-42
- [12] Ballabio D, Consonni V, Todeschini R 2008 The Kohonen and CP-ANN toolbox: A collection of MATLAB modules for Self Organizing Maps and Counterpropagation Artificial Neural Networks *Chemometrics and Intelligent Laboratory Systems* **98** 115-22

Authors	
	<p>Hongwei Ye, born in March, 1979, Heyuan City, Guangdong Province, P.R. China</p> <p>Current position, grades: the lecture of School of Electronic and Information, Heyuan Polytechnic, China. University studies: MSc in Computer Software And Theory from Sun Yat-sen University in China. Scientific interest: computer software, and neural network. Publications: 11 papers. Experience: teaching experience of 11 years, 2 scientific research projects.</p>
	<p>Lianjiao Zhang, born in October, 1980, Heyuan City, Guangdong Province, P.R. China</p> <p>Current position, grades: the lecture of School of Electronic and Information, Heyuan Polytechnic, China. University studies: MSc from Huazhong University of Science and Technology in China. Scientific interest: information engineering. Publications: 6 papers. Experience: teaching experience of 11 years, 3 scientific research projects.</p>
	<p>Xiaozhang Liu, born in November, 1978, Dongguan City, Guangdong Province, P.R. China</p> <p>Current position, grades: the associate professor of School of Computer Science, Dongguan University of Technology, China. University studies: PhD in Computational Science from Sun Yat-sen University in China. Scientific interest: pattern recognition, and wireless sensor networks. Publications: 20 papers. Experience: teaching experience of 10 years, 2 scientific research projects.</p>