International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.303-314
http://dx.doi.org/10.14257/ijsia.2014.8.6.27

A New Factor State Space Model for SCADA Network Attack and Defense

Li Yang

School of Computer Science, Southwest Petroleum University, Chengdu, China
Ling Wang, Xinyu Geng, Xiedong Cao
Southwest Petroleum University, Chengdu, China
scxdyl@126.com

Abstract

To solve the security problem in the supervisor control and data acquisition (SCADA), a new factor network model of SCADA network attack and defense based on factor state space is presented. Combining with factor space theory, formal descriptions of factor neurons based on factor state space are developed. On the basis of analysis and expression of network attack and defense factors, factor neuron model based on variable weight is proposed and a FNN-based security defense architecture model for SCADA network is put forward. For illustration, by introducing factor space canes, an attack simulation experiment is utilized to show the feasibility of the proposed method in solving network attack and defense knowledge reasoning. Experimental results indicate that the proposed method can effectively improve recognition rate of different attacks. Factor neuron network based on factor state space can effectively solve complexity of knowledge reasoning and expression in network attack and defense system and provides a new method for solving similar application.

Keywords: *SCADA, factor space, FNN, knowledge database*

1. Introduction

Supervisor control and data acquisition (SCADA) systems are widely used in various fields of electric power, water, petroleum, chemical industry, metallurgy, transportation *etc*. Once the systems have security vulnerability, Industrial production and national economic security will be faced with serious threats [1-2]. In the past, SCADA systems were regarded as a relative isolate, safe system which has strong access control ability. Nowadays, Now a lot of SCADA system has been connected to the Internet enterprises, which become relatively open and transparent, which makes the SCADA system faced with many security problems, such as malicious virus, information leakage and tampering, and *etc.*, In recent years, attacks on industry control system are becoming more common [3-4]. For example: in 2010, "the network super weapon" Stuxnet virus went through the invasion of ICS system, and threatened seriously to the safe operation of Iran Bushehr nuclear power plant and nuclear reactor; In 2011, hackers entered supervisory control and data acquisition system SCADA, so that the water supply pumps of USA Illinois city water supply system were destroyed [5-6]. Therefore, strengthening the security of SCADA system has become a problem that can not be neglected in industrial control, security problem of SCADA system has been attached more and more attention.

With requirement analysis of attack and defense of SCADA system, attack and defense system of SCADA can be taken as online digital intelligent antagonizing

ISSN: 1738-9976 IJSIA

process and all reasoning judgment, thinking and expression in attacking and defense. In this paper, with introduction of factor state space, analysis factor neuron and analog factor neuron, a new FNN-based SCADA network security defense architecture is presented to provide base frame for building SCADA defense system.

The remainder of the paper is organized as follows. Section 2, 3, 4 describe the formal description model of factor neural network, including factor state space, analysis factor neuron and analog factor neuron. And experimental result is discussed in section 5. Finally, we conclude our paper in Section 6, and provide suggestions for future work.

2. Formal Description of Factor State space

2.1. Factor and Factor Space

Factors are basic elements which can describe objects, such as properties of objects and conditions in rules [7-8].The meaning of the factor can be understood from the following three aspects:

(1)Attribution: it has two meanings. The first is that when looking for reasons from the results, factors are defined as objects cause some results. The second is that while the name is selected by state or feature, the factors will be symbols of a kind of state or a set of features [9-10].

(2)Analyticity: factors can be understood as a way to resolve the real world, an object can be described from different aspects in a different way, and the analysis process is the process of looking for factors. Factor is common character of an object, such as age, height, profession of a person.

(3)Descriptive: everything is the intersection of the various factors, a person can be identified from age, sex, height, weight, profession, educational history and etc., and a person is an intersection of the factors. And this means that it can build a broad cross-coordinate system, such objects can be described as a point of the generalized coordinates, and factor is the name of the dimension of the generalized coordinates.

2.2. Factors State Space Based on Object

The object u is related to factor f, which can view the object u from the point of f, and also there is a correspondingly state $f(u)$ associated with it. If U and F are the sets comprising some objects and some factors, and for any $u \in U$, all the factors related to U are in F $(f \in F)$. For a practical problem, we can always assume that there is an approximate matching. For a given matching (u, f), a correlation, R, between u and f, is defined, written as $R(u, f)$.Only when $R(u, f) = 1$, f and u are relevant. So u space related to f and f space related to u, respectively, can be defined as:

$$D(f) = \{u \in U \mid R(u, f) = 1\}$$

$$F(u) = \{f \in F \mid R(u, f) = 1\}$$

Factor $f(f \in F)$ can be regarded as a mapping and function in a certain object $u(u \in U)$ to access to certain state $f(u)$. $f : D(f) \to X(f)$, among them, $X(f) = \{f(u) \mid u \in U\}$, $X(f)$ is the state space of the f [11-12].

2.3. Knowledge Factor Expression

[Definition 1] In the domain of U, the atomic model of knowledge factors is a triple,

$$M(o) = < o, F, X >$$

Where ○ is a set of objects of the knowledge description about *U*.
F is a factor set when U is used to describe ○.
X is a state set about F when F is used to describe ○, and

$$X = \{X_O(f) \mid f \in F, o \in O\}$$

[Definition 2] In the domain of U, the relation of knowledge mode is defined as

$$R(O) = < RM, M(O), XM >$$

Where RM is a knowledge model.

$M(O)$ is atomic model of knowledge representation in knowledge model.

XM is structure group state and state transformation relation of the atomic model $M(O)$ in RM.

The atomic model of the knowledge factor representation gives a discrete set that describes objects; this is the basis of knowledge representation with factors. The relation mode of knowledge factor representation can associate with various related knowledge or different knowledge representation; this can realize the transformation of the different ways of knowledge and knowledge reasoning. They provide the basis of representation and processing of knowledge in using factors neural network.

2.4. Formal Description of Analysis Factor neuron

An analysis factor neuron model can be described as follows:

M= {<<O,G>,F,X>,<P,Q,R>,<A,B>}

Where O is a set of objects in the network system; G is the structure relation in the network; F represents cognition and description factor sets is state space of factor set in the network; O,G,F,X determine the state and structure of the system together; P,Q,R is respectively reasoning, judgment and control rule set. They together complete main independent operations and control functions;

A is input from outside and B is the target or response of information processing.

As a network consists of many neurons, an analysis factor neuron with reasoning function can be rewritten as:

$$M_i = \{< G_i, F_i, X_i >, < p, q, r >, < a, b >\}$$

Where $< G_i, F_i, X_i >$ together describes the structure, factor and states of factor neuron.

p,q,r respectively implements the reasoning, judgment and inner control function of factor neuron; a is input information; b is the target or response of factor neuron reasoning.

2.5. Formal Description of Analog Factor Neuron

As shown in Figure 1, in the network, there is a controllable series-parallel connection network which consists of many mini-cells. $f_1, \ldots, f_m$ are input factors relate to o, each factor is called a perceptible channel of analog factor neuron. $g_1, \ldots, g_n$ are output factors relate to *o*, they represent different output response.

Let $F_o = \{f_1, f_2, \ldots, f_m\}$

$$G_o = \{g_1, g_2, \ldots, g_m\}$$

$$X_o(F_o) = \{X_o(f_i) \mid i = 1, \ldots, m\}$$

$$Y_o(G_o) = \{X_o(g_j) \mid j = 1, \ldots, m\}$$

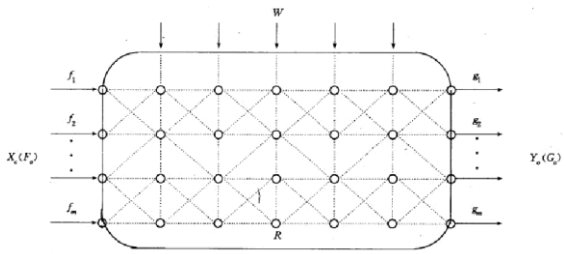


Figure 1. Analog Factor Neural Network Structure

For analog factors neuron, its external function can be expressed as:

$$Y_o(G_o) = R(X_o(F_o))$$

2.6. Formal Description of Combined Factor Neurons

If combined neural network system M is composed of N independent units or subsystems, then

$$M = \{M_i\}(i = 1, 2, \ldots N)$$

Let the state of subsystem M_i of a combined factor neural network be X_i, function state *X* of the system can be made of the various subsystems function state set $\{X_i\}$ vector, that is to say:

$$X = \{X_1, X_2, X_3, \ldots, X_N\}$$

A general order of factor neural network system can be realized by the weighted and controllable connections of various subsystems, Let $R = \{r_{ij}\}$ represent the relationship between the various subsystems, then:

$$r_{ij} = W_{ij} \bullet e_{ij}$$

Where e_{ij} is the connection relationship on function and structure between subsystems. *Wij* is transformation parameters of controllable function connection in the system.

Combined factor neural network subsystems set $\{M_{ij}\}$, subsystem state set $\{X_{ij}\}$ and controllable connection $R = \{r_{ij}\}$ together make up system structure.

The system function realization is reflected by system dynamic equation.

$$U = F(t, X(t), I(t))$$

Where $X(t)$ represents the system's overall state; $I(t)$ describes system outside input states; F() for mapping function of changes state of the system.

When N systems make up a factor neural network system, each subsystem state function dynamic equation is expressed as:

$$U_i = h_i(t, X_i, I_i + \sum g_{ij}(t, X_i, I_j)(i = 1, 2, \ldots, N)$$

Where X_i, X_j is respectively the current state of the subsystem i and j;

I_i, I_j is respectively the current input of the subsystem i and j;

h_i *is the feedback state change mapping function of the subsystem i;* g_{ij} *is state change effect mapping function from subsystem j to subsystem i.*

3. Factor Neuron Network Model based on Factor Space

Given that an atom factor space $\{X(f)\}_{(f \in F)}$ and atom factors cluster $\pi = \{f_1, f_2, ..., f_m\} \subset F$, for an object u, the state of u can be written as $x_j = f_j(u)(j = 1, 2, ..., m)$, then we can obtain the state of u in full space $X(1)$:

$$x = 1(u) \approx M_m(f_1(u), f_2(u), ..., f_m(u)) = M_m(x_1, x_2, ..., x_m) \tag{1}$$

Where $M_m : [0,1]^m \to [0,1]$ is an integrating function; especially, we can assume that $M_m = \Sigma$, that is to say,

$$x = M_m(x_1, x_2, ..., x_m) = \sum_{j=1}^{m} w_j x_j \tag{2}$$

Where $w_j (j = 1, 2, ..., m)$ is a constant weight value:

$$w_j \in [0,1](j = 1, 2, ..., m) \text{ and } \sum_{j=1}^{m} w_j = 1$$

As is shown in Figure 2, factor space can be regarded as a converter, which can receive a set of state variables $(x_1, x_2, ..., x_m)$ and integrate the variables and get an output variable $x = M_m(x_1, x_2, ..., x_m)$.

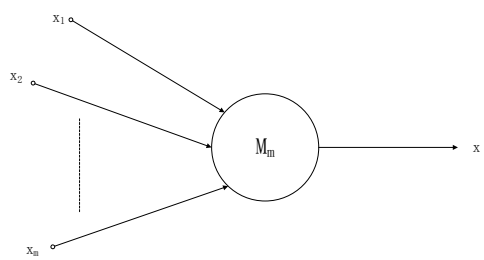


Figure 2. Integration Function of Factor Space

When $M_m = \Sigma$, atom factors $f_1, f_2, ..., f_m$ can be regarded as m input channels, $\omega_1, \omega_2, ..., \omega_m$ is respectively damping coefficient, full factor 1 is looked as output channel. As shown in Figure 3, a neuron model can receive a set of input variables $(x_1, x_2, ..., x_m)$ and get an output variable $x = \sum_{j=1}^{m} \omega_j x_j$.

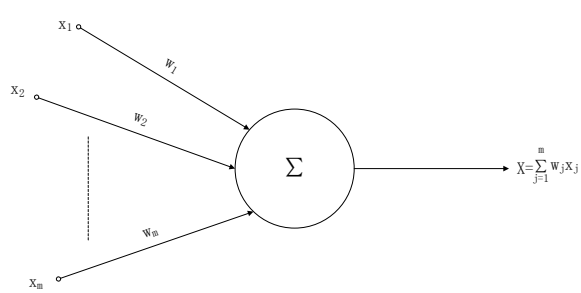


Figure 3. Neuron Model based on Factor Space

3.1. Threshold Neuron Model Unrelated to Time

As illustrated in Figure 4, threshold $\theta \in [0,1]$ is added to output channel. When integrated result exceeds the value θ, the result can be output, or the result is zero. The output y of the neuron can be expressed as the following:

$$y = \varphi(\sum_{j=1}^{m} \omega_j x_j - \theta) \tag{3}$$

Where $\varphi(x)$ is a segmented function, which is defined as

$$\varphi(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{4}$$

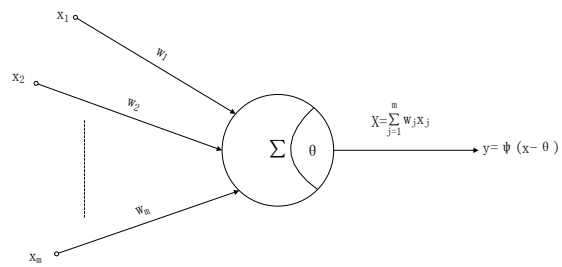


Figure 4. Neuron Model with Threshold Value

3.2. Linear Neuron Model Unrelated to Time

When threshold $\theta = 0$, ”half-wave rectification” function φ makes failure. Equation (3) can be simplified to linear function

$$x = \sum_{j=1}^{m} \omega_j x_j$$

which is a linear neuron model and special case of threshold neuron model.

3.2. General Expression of Linear Neuron Model Unrelated to Time

In equation (3), we can convert function $\sum$ to general function M_m, thus we can obtain the following general expression of threshold neuron model:

$$y = \varphi(M_m(x_1, x_2, \ldots, x_m) - \theta) \tag{5}$$

When $\theta = 0$, equation (5) can be simplified to (6)

$$x = M_m(x_1, x_2, \ldots, x_m) \tag{6}$$

According to different form of M_m, (6) has different special case:

$$x = M_m(x_1, x_2, \ldots, x_m) = \bigwedge_{j=1}^{m} x_j \tag{7}$$

$$x = M_m(x_1, x_2, \ldots, x_m) = \bigvee_{j=1}^{m} x_j \tag{8}$$

$$x = M_m(x_1, x_2, \ldots, x_m) = \sum_{j=1}^{m} \omega_j x_j \tag{9}$$

Where $\omega_j \in [0,1](j = 1, 2, \ldots, m)$ and $\sum_{j=1}^{m} \omega_j = 1$;

$$x = \sum_{j=1}^{m} \omega_j(x_j) x_j \tag{10}$$

Where $\omega_j:[0,1]\to[0,1]$, $t\mapsto\omega_j(t)$ is continuous function and satisfies normalizing condition: $\sum_{j=1}^{m}\omega_j(x_j)=1$, $\omega_j(x_j)$ is an univariate weight;

$$x=\bigvee_{j=1}^{m}(\omega_j x_j) \tag{11}$$

Where $\omega_j\in[0,1](j=1,2,\ldots,m)$ and $\bigvee_{j=1}^{m}\omega_j=1$;

$$x=\bigvee_{j=1}^{m}(\omega_j(x_j)x_j) \tag{12}$$

Where $\omega_j(x_j)$ is the same as (10), its normalizing condition is as follows:

$$\bigvee_{j=1}^{m}\omega_j(x_j)=1 \tag{13}$$

$$x=\bigvee_{j=1}^{m}(\omega_j\wedge x_j) \tag{14}$$

Where $\omega_j\in[0,1](j=1,2,\ldots,m)$ and $\bigvee_{j=1}^{m}\omega_j=1$;

$$x=\bigvee_{j=1}^{m}(\omega_j(x_j)\wedge x_j) \tag{15}$$

Where $\omega_j(x_j)$ is a univariate weight which satisfies equation (13);

$$x=(\prod_{j=1}^{m}x_j)^{\frac{1}{m}} \tag{16}$$

$$x=(\frac{1}{m}\sum_{j=1}^{m}x_j^p)^{\frac{1}{m}},\ p>0 \tag{17}$$

$$x=(\sum_{j=1}^{m}\omega_j x_j^p)^{\frac{1}{p}} \tag{18}$$

Where $p>0,\omega_j\in[0,1](j=1,2,\ldots,m)$ and $\sum_{j=1}^{m}\omega_j=1$;

$$x=(\sum_{j=1}^{m}\omega_j(x_j)x_j^p)^{\frac{1}{p}} \tag{19}$$

Where $p>0$, $\omega_j(x_j)$ is a univariate weight which satisfies $\sum_{j=1}^{m}\omega_j(x_j)=1$;

The above models can generate more complex neuron models by integrating functions.

4. Factor Neuron Model based on Variable Weight

As is mentioned above, under the condition of constant weight, if negative weight is not introduced, excitation and inhibition mechanism can not be implemented. But if weight value ω_j changes with the change of x_j, excitation and inhibition mechanism can be implemented with the change of weight value.

[Definition 3] A set of function of one variable $\omega_j:[0,1]\to[0,1](j=1,2,\ldots,m)$ are called a set of univariate weight of one variable if each $\omega_j(x)$ is monotone and continuous function. If $\omega_j(x)$ is monotone increasing function, $\omega_j(x)$ is called excitation weight; If $\omega_j(x)$ is monotone decreasing function, $\omega_j(x)$ is called inhibition weight.

It is noted that univariate weight of one variable defined above doesn't necessarily satisfy normalizing condition:

$$\sum_{j=1}^{m} \omega_j(x_j) = 1 .$$

Assume that $\omega_1(x),\ldots,\omega_p(x)$ are excitation weights , $\omega_{p+1}(x),\ldots,\omega_m(x)$ are inhibition weights, and

$$e = \sum_{j=1}^{p} \omega_j(x_j)x_j , \ h = \sum_{j=p+1}^{m} \omega_j(x_j)x_j$$

We can obtain Fukushima neuron model:

$$y = \varphi((\varepsilon + e)/(\varepsilon + h) - 1) = \varphi((e - h)/(\varepsilon + h)) \tag{20}$$

In fact, input intensity x_j changes with time, that is to say, $x_j = x_j(t)$, then ω_j changes with time,

$\omega_j = \omega_j(x_j(t))$, so we can get the following formula:

$$y(t+\tau) = \varphi(\sum_{j=1}^{m} \omega_j(x_j(t))x_j(t) - \theta) \tag{21}$$

In addition, let

$$e(t) = \sum_{j=1}^{p} \omega_j(x_j(t))x_j(t), h(t) = \sum_{j=p+1}^{m} \omega_j(x_j(t))x_j(t) \tag{22}$$

We can obtain Fukushima neuron model with time variable.

$$y(t+\tau) = \varphi((e(t) - h(t))/(\varepsilon + h(t))) \tag{23}$$

5. Simulation Experiment on SCADA Network Attack and Defense

We collected 494,021 records by tcpdump tool. Each record contains 41 feature values and 1 attack type description. With the knowledge representation theory of factor space, we got the following attack type factor space set. O={all attacking behaviors},F={data link feature set, the attacking type}={F1, F2}, F2=abnormal behavior type, for F2 is able to generate G={PROBE, DOS, U2R, R2L}, so you can build the following factors space canes according to factor space canes theory:

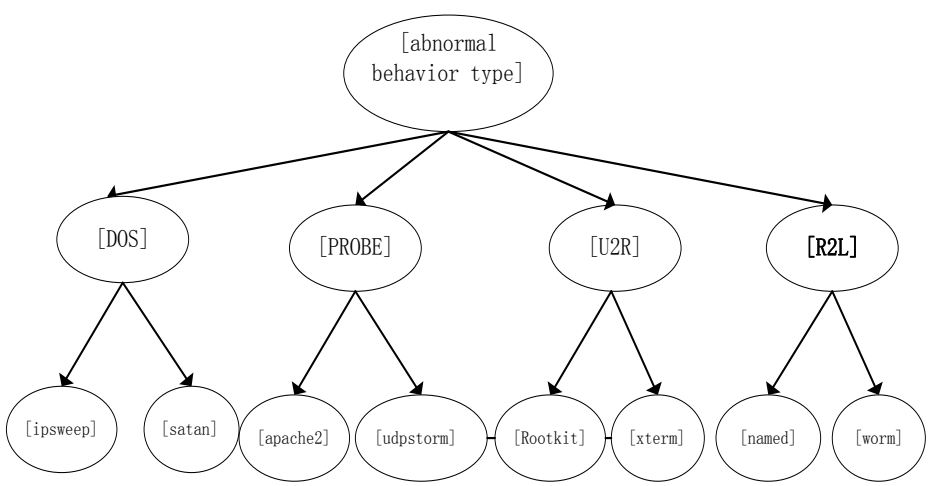


Figure 5. Factors Space Canes

Attack type factor space set is shown in Table 1.

Table 1. Attack Type Factor Space Set

Node number	Type	set code	Parent node
0	all attacking behaviors	O	-1
1	data link feature set	F1	0
2	the attacking type	F2	0
3	DOS	F21	2
4	PROBE	F22	2
5	U2R	F23	2
6	R2L	F24	2
7	TCP link basic features,	F11	1
8	TCP link content features	F12	1
9	time-based network traffic statistical features	F13	1
10	host-based network traffic statistical features	F14	1
11	duration	G11	7
12	protocol type	G12	7
13	service	G13	7

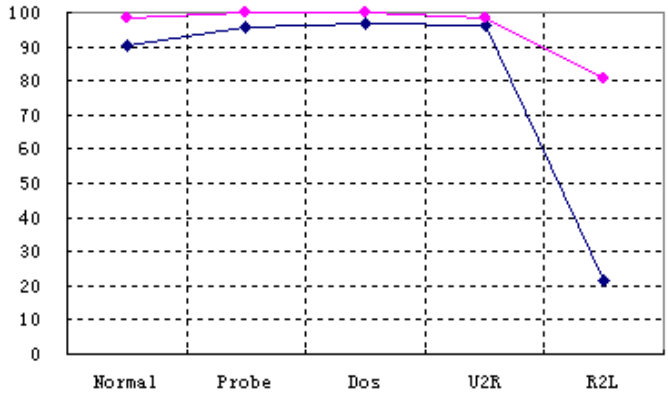
If F2=[abnormal type]= $\{X(f)\}f \in F2$. F21=[DOS]\[abnormal type], F22=[PROBE]\[abnormal type], F23=[U2R]\[abnormal type], F24=[R2L]\[abnormal type], then [DOS]= $\{X(f)\}f \in F21$, [PROBE]= $\{X(f)\}f \in F22$, [U2R]= $\{X(f)\}f \in F23$, [R2L]= $\{X(f)\}f \in F24$.

F1 is a link factor set, and F1={TCP link basic features, TCP link content features , time-based network traffic statistical features, host-based network traffic statistical features}={F11, F12, F13, F14}, F11can generate 9 seed factors, such as G11={duration, protocol type, service,…} etc; F12 can generate 13 seed factors, F13 can produce 9 seed factors, F14 can generate 10 seed factors. Similarly, it can produce melon factor space canes.

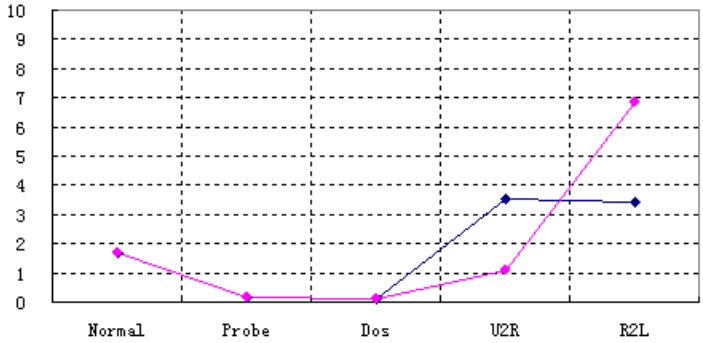
Therefore, we can establish reasoning rules from F1 link factors to F2 abnormal type factors, these rules have shown the corresponding relationship between connection factor state vector sets and abnormal type state vector sets, in addition, on the basis of this, we can establish deduction matrix to generate $X_{F1} \square R \rightarrow X_{F2}$.

When the system inputted a set of connection state X'_{F1}, according to R, the computer can infer as follows: $X'_{F1} \square R \rightarrow X'_{F2}$.

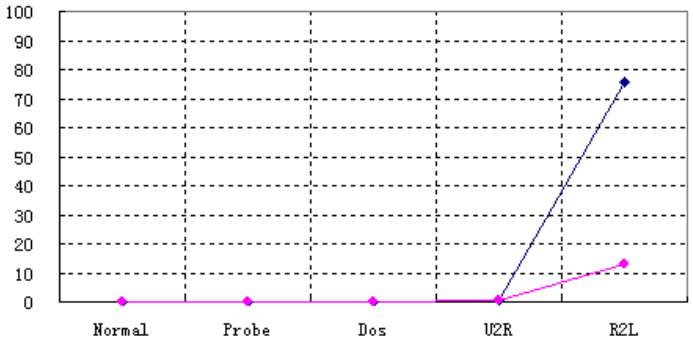
In the simulation experiment, neural network has 9 input nodes, and 9 connection factors. Hidden layer uses 40 nodes, output layer uses 1 nodes. Initial learning rate is set to 0.3, by training 200 times. As shown in Figure 6, “red color” corresponds to factor neuron method, ”blue color” corresponds to analog neuron method. From Figure 4 we can see: Since percentage of PROBE and DOS in the database is high, recognition rate of PROBE and DOS is high and as percentage of R2L in the database is low, recognition rate of factor neural is higher than analog neuron. False rate PROBE and DOS are relatively low, there is little different in recognition rate between factor neuron and analog neuron. As a result, in general, our proposed model has improved detecting efficiency, reduced missing rate and false rate.



(a). Recognition rate of analog neuron and factor analog neuron



(b). False rate of analog neuron and factor analog neuron



(c). Missing rate of analog neuron and factor analog neuron

Figure 6. Detection Rate Comparison of Analog Neuron with Factor Neuron under Different type Attacks

"red color" corresponds to factor neuron method, "blue color" corresponds to analog neuron method

6. Conclusions

Analyzing factor, factor space, attack factors, skill and attack aim, this paper gives a new knowledge factor expression model of SCADA network attack and defense system, presents factor neuron model unrelated to time based on factor space of network attack

and defense and proposes factor neuron model based on variable weight The experimental results further verify the valid of our proposed method, which has a higher true recognition rate than other methods. Factor neuron model based on factor state space is a new modeling approach which can be used in factor analysis and expression of network attack and other fields. We will make research on a new factor space reasoning model and discuss equivalence partitioning and associating mechanisms in an analysis factor neuron in the next work.

Acknowledgements

This work was supported by National Natural Science Foundation Project under grants 61175122, as well as by Applied Basic Research Project of Sichuan province of China (2013JY0134).

References

[1] M. De Vivo, G. O. de Vivo and G. Isern, “Internet security attacks at the basic levels”, Operating Systems Review, vol. 32, no. 2, (**2002**), pp. 40-48.
[2] L. Teo, Y. Zheng and G. Ahn, “Intrusion detection force: an infrastructure for internet-scale intrusion detection”, Proceedings of the First IEEE International Workshop on Information Assurance (IWIA’03), Darmstadt, Germany, (**2003**), pp. 73-91.
[3] Z, Feng and X, Zhao, “American Network centric warfare”, Beijing: National Defense University Press, China, (**2004**).
[4] L. Zheng, Z. Liu and Y. Wu, “Network warfare on the battlefield”, Beijing: military science press, China, (**2002**).
[5] Z. Liu and Y. Liu, “Factor neural network theory and implementation strategy research”, Beijing: Beijing Normal University Press, China, (**1992**).
[6] X. Cao, *et al.*, “The Geological Disasters Defense Expert System of the Massive Pipeline Network SCADA System Based on FNN”, Lecture Notes in Computer Science, Springer press, vol. 1, no. 7234, (**2012**), pp. 19-26.
[7] L. Yang, *et al.*, “A New Formal Description Model of Network Attacking and Defense Knowledge of Oil and Gas Field SCADA System”, Lecture Notes in Computer Science, Springer press, vol. 1, no. 7234, (**2012**), pp. 2-10.
[8] G. Chun-xia, L. Zeng-liang and M. Qing, “Network attack planning model and its generating algorithm”, Computer Engineering and Applications, vol. 46, no. 31, (**2010**), pp. 121-123.
[9] T. Zhang, “Study on the network attack prediction algorithm based on chaos theory and LSSVM”, Journal of Convergence Information Technology, vol. 7, no. 11, (**2012**), pp. 224-231.
[10] T. Li and N. Xiao, “An Approach to Feature Dimensionality Reduction Based on Radial Basis Function Neural Networks”, Journal of Convergence Information Technology, vol. 7, no. 3, (**2012**), pp. 117-126.
[11] 1. H. Debar, M. Dacier and A. Wespi, "Towards a taxonomy of intrusion detection systems”, Computer Networks, vol. 31, no. 8, (**1999**) April, pp. 805-822.
[12] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines”, In Proceedings of the International Joint Conference on Neural Networks, (**2003**) July, pp. 1741-1745.
[13] Y. Liu, D. Tian and D. Wei, "A wireless intrusion detection method based on neural network”, in Proceedings of the 2nd IASTED international conference on Advances in computer science and technology, (**2006**) January, pp. 207-211, Puerto Vallarta, Mexico.
[14] V. Dao and V. Vemuri, "A performance comparison of different back propagation neural networks methods in computer network intrusion detection”, Differential of Equations and dynamical Systems, (**2002**) January and April, pp. 201-214.
[15] C. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, no. 2, (**1998**), pp. 121-167.
[16] 10. B. Scho?lkopf, R. Williamson, *et al.*, "Support vector method for novelty detection", in Neural Information processing Systems, MIT Press, (**2000**), pp. 582-588.
[17] T. Tsiakis, "Information Security Expenditures: a Techno-Economic Analysis", International Journal of Computer Science and Network Security, vol. 10.4, (**2010**) April, pp. 7-11.
[18] T. Peng, C. Leckie and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems", ACM Computing Surveys, vol. 39, no. 1, Article 3, (**2007**) April.

Authors

Li Yang, he received his M.S. degree in computer science and technology from Southwest Petroleum University, Sichuan, China, in 2006. He is a research scientist in School of Computer Science, Southwest Petroleum University, Chengdu, China. His main research interest is artificial intelligence, distributed computing and network security.

Ling wang, she received B.S. and M.S. degree in mathematics in 1987 and in 1990, respectively. He received the Ph.D. degree from Xidian University in 2000. He is currently a professor in School of Computer Science, Southwest Petroleum University, China. Her current research interests include algebra, information security and image processing, pattern recognition and their applications.

Xinyu Geng, he received M.S. degree from Chongqing University, Chongqing, China, in 1984. He has been a full professor in School of Computer Science, Southwest Petroleum University, Chengdu, China. His main research interests include neural network, data mining. In particular, he is interested in designing and analyzing algorithms for many computationally hard problems in computer networks.

Xiedong Cao, he received an M.S. degree from Southwest Petroleum University, Chengdu, China, in 1984. He is now a professor in Southwest Petroleum University, director of the Computer Detection and Control Laboratory, academic leader of provincial key discipline and a senior member of China Computer Federation. His research interests cover intelligent SCADA, Fuzzy networks and intelligent control.

