https://doi.org/10.31449/inf.v46i1.3775 Informatica **46** (2022) 87–93 **87**

Network Security Situational Level Prediction Based on a Double-Feedback Elman Model

Jinbao He and Jie Yang
E-mail: jinbhe@yeah.net
Qian'an College, North China University of Science and Technology, Tangshan, Hebei 064400, China

Student paper

Keywords: network security, neural network, situational level, back-propagation neural network, situational prediction

Received: October 8, 2021

Network Security Situational Awareness (NSSA) is an important element in network security research. Predicting network security situational level can help grasp the network security situation. This study mainly focuses on the double-feedback Elman model. Firstly, NSSA was briefly introduced. Then, relevant indicators were selected to establish a security situational indicator system. A back-propagation neural network (BPNN) model was designed to evaluate the situational value. A dual-feedback Elman model was used to predict the future situational level. The actual network environment was built to conduct experiments. The results showed that the evaluation results of only three samples obtained by the BPNN model did not match the actual situation, with an accuracy of 90%, and the prediction results of only four samples obtained by the dual-feedback Elman model did not match the actual situation, with an accuracy of 96.67%. The experimental results verify the reliability of the network security situational level prediction method designed in this study. The NSSA method can be promoted and applied in practice.

Povzetek: S pomočjo globokih nevronskih mrež so razvili metodo za napovedovanje glede varnosti v omrežjih.

1 Introduction

As the network develops rapidly [1], people's living standards have gradually improved [2], and the growing scale of the network and the more and more complex network environment has led to a higher frequency of security problems, which greatly challenges the integrity and confidentiality of network data [3]. Although there are many security measures, such as firewalls [4] and anomaly detection [5], none of them can provide a systematic and holistic perception of the network, discover the problems and solve them, which leads to the emergence of Network Security Situational Awareness (NSSA) [6]. NSSA can fuse all available information to assess the situation of the network, provide network administrators with relevant decision bases, and minimize possible risks and losses, which is important for improving the monitoring capability and responsiveness of the network. NSSA has currently become a hot topic. Zhao and Liu [7] used a parallel reduction algorithm to reduce all data, optimized the wavelet neural network by the particle swarm algorithm, and performed situational awareness. The simulation experiments showed that the method had a higher convergence speed and better fitting effect. Zhang et al. [8] proposed a stochastic game-based method, quantified the situational value of the network with the utility of both sides of the game, and predicted the attack behavior by Nash equilibrium. The experiments found that the method could well reflect the change of network situation and predict the attack behavior. Li et al. [9] used Glowworm Swarm Optimization (GSO) to optimize the Gaussian process and performed situational prediction. They found through experiments that the method had better convergence and smaller prediction error. Hu et al. [10] used MapReduce for distributed training using a support vector machine (SVM) and built a situation prediction model. The experiment found that the method effectively improved accuracy and reduced time cost compared with the traditional method. In this study, NSSA was analyzed based on neural networks, the back-propagation neural network (BPNN) model was used for situational assessment, the dual-feedback Elman model was used for situational prediction, and an experimental analysis was conducted to verify the effectiveness of the method. This work provides a stronger guarantee for network security.

2 Network security situational awareness

Network security refers to protect computer systems from damage and also to avoid interruption of computer services. With the widespread use of computers, the Internet and WiFi in life, and the promotion of smart terminals and small smart devices, network security has become more important.

Situational awareness originated in military thinking and was first applied in aviation [11]. Then, it was also

well applied in medicine, electric power, and networks [12]. It mainly includes three stages. The first stage is understanding the situation, i.e., acquiring raw data through sensors, network monitoring, etc., and pre-processing them to extract useful security elements. The second stage is evaluating the situation, i.e., processing the extract security elements with fusion and association to evaluate the current network state. The third stage is predicting the situation, i.e., finding out the underlying laws through some prediction methods according to the historical situational value.

In the big data environment, the network behaves like a more massive and complex system, and at the same time, it also faces more security risks. The massive amount of data has certain errors and redundancy and stronger correlation and uncertainty [13] and changes faster, so the difficulty of data processing is further enhanced, which brings more difficulties to NSSA. Improving the efficiency of NSSA and the timeliness and accuracy of prediction to better achieve network security is an important issue at present. More and more methods have been applied in NSSA, such as machine learning, immune systems, game analysis and visualization techniques [14]. This study focuses on the neural network algorithm.

3 The double-feedback Elman model-based prediction model

3.1 Network security situational indicator system

Network security situational assessment is the basis of NSSA. The network data collected needs to be processed scientifically to guide administrators' decision-making. Firstly, relevant indicators should be selected to establish an indicator system that can comprehensively, objectively and scientifically reflect the network security situation. This study divided the security situation into four independent level 1 indicators and selected level 2 indicators to describe, as shown in Table 1.

Due to the complexity of the network environment, some of the indicators collected in Table 1 were nominal data that could not be directly input into the model for calculation; therefore, these indicators were quantified. Based on the Common Vulnerability Scoring System (CVSS) 3.0, these indicators were calculated as follows.

(1) Network vulnerability level: This indicator is mainly affected by the number and type of vulnerabilities, and its calculation formula is:

$$g = \frac{\sum_{i=1}^{n}\sum_{j=1}^{M} w_{ij}Q_i D_{ij}}{N},$$

$$Q_i = \frac{I_i}{\sum_{i=1}^{n} I_i},$$

$$I_i = \begin{cases} 1.0, & confidential \\ 0.7, & important \\ 0.4, & ordinary \end{cases},$$

where n refers to the number of hosts, M refers to the number of vulnerability categories, N refers to the total number of vulnerabilities, w_{ij} refers to the grade factor of the j-th vulnerability category in the i-th host, which can be obtained according to CVSS, D_{ij} refers to the number of the j-th vulnerability category in the i-th host, Q_i refers to the importance indicator of the i-th device, and I_i refers to the importance score of information scored by the host.

Level 1 indicator	Level 2 indicator
Stability indicators	Mean free error time
	Rate of change in flow
	Total data flow
	Number of surviving critical devices in the network
Threat indicators	Number of alarms
	Bandwidth utilization
	Data inflow
	Historical frequency of security incidents
Vulnerability indicators	Number of network vulnerabilities
	Network vulnerability level
	Total number of open ports for critical devices
	Network topology
Disaster tolerance indicators	Network bandwidth
	Number of safety equipment
	Host operating system
	Number of concurrent threads supported by the primary server

Table 1: Situational assessment indicator system.

Operating system	Versions	Score
Windows	XP	1
	7	2
	8	3
	10	4
	Server 2016	5
Ubuntu	14.4	4
	16.4	5
	Web Server 12	5
	Web Server 16	6
Mac OS	10	8

Table 2: Corresponding scores for operating systems.

(2) Network topology: Topology refers to the layout of security devices, and its calculation formula is:

$$g = \sum_{i=1}^{n} T_i,$$

$$T_i = \begin{cases} 1.0, m \in [0,3) \\ 0.5, m \in [3,5) \\ 0.1, m \in [5,+\infty) \end{cases},$$

where T_i stands for the security score of the i-th topology and m stands for the number of nodes in the topology.

(3) Host operating system: the higher the version of the operating system is, the smoother the operation of the network is. The points corresponding to different operating systems [15] are shown in Table 2, and the overall calculation formula is:

$$g = \sum_{i}^{N} \mathrm{OSTypeScore}_i,$$

where $OSTypeScore_i$ refers to the score of the operating system of the i-th device.

At the same time, the indicators were normalized, and the values were in $[0, 1]$. The corresponding formula is:

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}.$$

Referring to the National Computer Network Emergency Response Technical Team/Coordination Center of China, the security situation was divided into five different levels, as shown in Table 3.

The hierarchical matrices between different level 1 indicators and the security situation are shown in Table 4.

The security levels of level 1 indicators are shown in Table 5.

Security indicator value	Dangerous level
0.0-0.2	Safe
0.2-0.4	Mildly dangerous
0.4 - 0.75	Generally dangerous
0.75-0.9	Moderately dangerous
0.9-1.0	Highly dangerous

Table 3: Network security situation levels.

	Safe	Mildly dangerous	Generally dangerous	Generally dangerous	Highly dangerous
Stability indicators	High	High	Medium	Low	Low
Threat indicators	Low	Medium	High/ medium	High	High
Vulnerability indicators	Low	Low/ medium	High/ medium	Medium/high	High
Disaster tolerance indicators	High	High/ medium	Medium	Medium/low	Low

Table 4: The hierarchical matrices of level 1 indicators.

	Low	Medium	High
Stability indicators	0-0.3	0.3-0.7	0.7-1.0
Threat indicators	0-0.4	0.4-0.7	0.7-1.0
Vulnerability indicators	0-0.3	0.3-0.6	0.6-1.0
Disaster tolerance indicators	0-0.4	0.4-0.8	0.8-1.0

Table 5: The security level table of level 1 indicators.

3.2 Assessment methodology of the network security situation

For the situation assessment, BPNN was chosen as the model in this study [16]. BPNN has strong adaptive, self-organizing and learning abilities, and it has a good effect on the uncertainty of security elements. It has a high learning speed and a relatively simple modeling process. Therefore, it is well suited for evaluating security situations.

Level 2 indicators in Table 1 were used as the input of BPNN, and level 1 indicators were used as the output of BPNN. Then, the situation of the whole network was evaluated. $A_i(i = 1,2,\cdots,m)$ denotes the i-th level one indicator, $B_{ij}(i = 1,2,\cdots,m, j = 1,2,\cdots,n)$ denotes the j-th level two indicator corresponding to the i-th level one indicator, and W and V denote the weight from the input layer to the hidden layer and the weight from the hidden layer to the output layer. The BPNN-based evaluation model is written as:

$$A = f(B, W, V).$$

Situational level L is a function of m level one indicators, which can be written as:

$$L = f(A_1, A_2, \cdots, A_m).$$

The BPNN model has a three-layer structure. The number of nodes in the input layer was the number of level two indicators. The number of nodes in the output layer was the number of level 1 indicators, i.e., 1. The number of nodes in the hidden layer was determined by an empirical formula [17]:

$$l = \sqrt{n + m} + a,$$

where n, m, and l are the number of nodes in the output, output, and hidden layer, respectively, and a is a regulation constant between 1 and 10. After determining the range of the nodes in the hidden layer, the final number of nodes in the hidden layer was determined by training networks containing different number of nodes in the hidden layer.

The model used the Sigmoid function as the activation function. The learning rate was 0.1. The maximum number of training was 1000.

	CPU	Memory	Hard disk	Network card	Operating system	Main services
Experimental machine 1	Inter(R)Core(TM) i5	4G	1T	100 M	Windows 7	DHCP, FTP, TELNET, HTTP
Experimental machine 2	Inter(R)Core(TM) i3	2G	1T	100 M	Windows XP	FTP, RPC, DNS
Experimental machine 3	Inter(R)Core(TM) i5	4G	1T	100 M	Windows 8	DHCP, FTP, TELNET, HTTP, Browser
Experimental machine 4	Inter(R)Core(TM) i5	4G	1T	100 M	Windows 10	DHCP, NLA, COM+, Event System, DCOM
Experimental machine 5	Inter(R)Atom(TM) N450	4G	1T	Gigabit	Ubuntu14	Sever, ICS, FTP

Table 6: Equipment configuration.

Sample number	The input of BPNN				Expert score	Evaluation result
1	0.59	0.67	0.81	0.83	0.78	High
2	0.24	0.26	0.28	0.25	0.25	Low
3	0.78	0.86	0.79	0.64	0.81	High
4	0.56	0.52	0.61	0.63	0.64	Medium
5	0.45	0.57	0.58	0.64	0.59	Medium
...	...	...	...	...	...	...
100	0.11	0.27	0.18	0.16	0.21	Low

Table 7: Experimental data for stability assessment.

3.3 Prediction method of network security situational level

Based on the security situational values obtained from the above evaluation, the future situation can be predicted.

The Elman neural network has the nonlinear dynamic characteristic and is is sensitive to time series data; therefore, the prediction of the network security situational level with the ELman neural network is a dynamic time series problem. Therefore, the Elman neural network [18] was improved to obtain a dual-feedback Elman neural network model for predicting security situation. Compared with the original Elman neural network, the dual-feedback Elman neural network achieved the timely correction of errors by increasing the feedback, which improved the learning speed of the model. The relevant equations are:

$$\mathrm{x(k)} = \mathrm{f}\left(\mathrm{W^{I1}x_c(k)} + \mathrm{W^{I2}u(k-1)} + \mathrm{W^{I4}y_c(k-1)}\right),$$

$$x_c(k) = a\left(x_c(k-1) + x(k-1)\right),$$

$$y_c(k) = \gamma\left(y_c(k-1) + y(k-1)\right),$$

$$\mathrm{y(k)} = \mathrm{g}\left(\mathrm{W^{I3}x(k)}\right),$$

where x(k) represents the output of the hidden layer, $\mathrm{x_c(k)}$ represents the output of unit 1 in the succession layer, $\mathrm{y_c(k)}$ represents the output of unit 2 in the succession layer, y(k) represents the output of the output layer, a and γ are adjustment factors, $\mathrm{W^{I1}}$, $\mathrm{W^{I2}}$, $\mathrm{W^{I3}}$ and $\mathrm{W^{I4}}$ are connection weights, and f(x) represents the transfer function, $\mathrm{f(x)} = \frac{1}{1+\mathrm{e^{-x}}}$.

Through the gradient descent algorithm, the partial derivatives of all the connection weights were calculated based on error E. It was assumed that the expected output was $\mathrm{y_d(k)}$, then its error function, i.e., the objective function, is written as:

$$\mathrm{E(k)} = \frac{1}{2}\left(\mathrm{y_d(k) - y(k)^T}\left(\mathrm{y_d(k) - y(k)}\right)\right).$$

4 Experimental analysis

Two desktop computers, two laptops and one cloud server were prepared to build the experimental environment. Two actual websites in the campus network were used as data sources. One of the desktops was used as the client, and the other four computers were used as the attackers. The data was MySQL5.1. The configurations of different devices are shown in Table 6.

The data were collected in the laboratory for ten consecutive days. Expert opinions were obtained anonymously. One hundred samples were generated for every level one indicator in Table 1. Among the 100 samples, 70 samples were randomly selected as the training set and 30 as the test set. Taking the stability evaluation as an example, the samples obtained are shown in Table 7.

Sample number	Model evaluation results	Actual results
1	Low	Medium
2	Medium	Medium
3	Medium	Medium
4	Medium	Medium
5	High	High
6	High	High
7	High	High
8	Medium	Medium
9	Medium	Medium
10	Medium	Medium
11	Low	Low
12	Medium	Medium
13	Low	Medium
14	High	High
15	High	High
16	Medium	Medium
17	Low	Low
18	High	High
19	Low	Low
20	Low	Medium
21	Medium	Medium
22	High	High
23	Low	Low
24	High	High
25	Medium	Medium
26	Medium	Medium
27	Low	Low
28	High	High
29	Medium	Medium
30	Medium	Medium

Table 8: Comparison of stability assessment.

Experiments were conducted on 30 test samples using the trained BPNN model, and the obtained results are shown in Figure 1.

It was seen from Figure 1 that the output of the BPNN model had a good match with the actual situation, indicating that the BPNN model could make a good evaluation of the situational values. The stability level corresponding to the output of the BPNN model was compared with the stability level corresponding to the actual situation, and the results are shown in Table 8.

It was found from Table 8 that only 3 out of the 30 test samples had inconsistent evaluation results with the actual situation. The evaluation result of sample 1 was "low", but its actual result was "medium", and the same was true for samples 13 and 20. The accuracy of the BPNN model for the assessment of stability situation reached 90%, which verified the reliability of the BPNN model. In conclusion, the BPNN model could make a more accurate assessment of the network security situation, which was conducive to the next situational prediction.

Five hundred samples were selected from the situation database for the experiment of situation prediction. The first five samples were used to predict the fifth sample, and so on. Five samples were regarded as one group, and there were 100 groups. Seventy groups were randomly selected as training samples, and 30 groups were used as test samples. The inputs and outputs of the model are shown in Table 9.

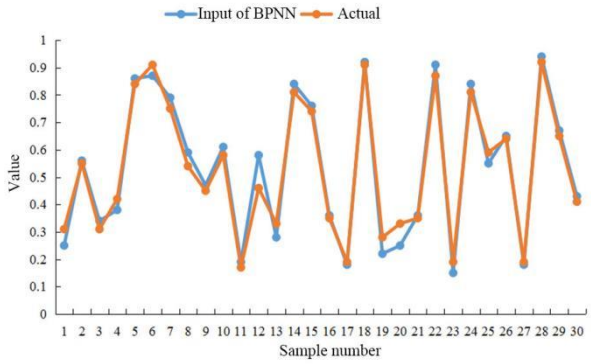

Figure 1: Results of the situational assessment.

Sample number	The first sample value	The second sample value	The third sample value	The fourth sample value	Sample Output
1	0.75	0.26	0.37	0.21	0.26
2	0.46	0.52	0.34	0.29	0.19
3	0.78	0.12	0.61	0.35	0.42
4	0.64	0.25	0.36	0.81	0.28
5	0.77	0.14	0.62	0.37	0.75
....	...	...	...	...	...
100	0.66	0.25	0.84	0.16	0.82

Table 9: Inputs and outputs of the model.

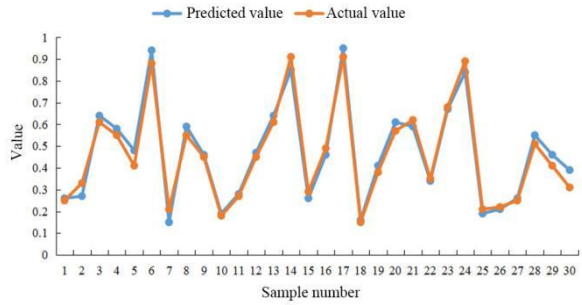

Figure 2: Prediction results of the situation.

The situation was predicted using the trained dual-feedback Elman model, and the results are shown in Figure 2.

It was seen from Figure 2 that the prediction results of the dual-feedback Elman model were closer to the actual values, and the predicted situational values matched well with the actual situation, indicating that the dual-feedback Elman model could make more accurate predictions of the situational values and the predicted results had high accuracy. Then, the predicted security situational levels were compared with the actual situation, and the results are shown in Table 10.

It was seen from Table 10 that four samples were predicted wrongly. The prediction result of sample 1 was highly dangerous, while it was moderately dangerous in fact. The prediction result of sample 14 was moderately dangerous, while it was highly dangerous in fact. The prediction result of sample 19 was generally dangerous, while it was mildly dangerous in fact. The prediction result of sample 25 was safe, while it was mildly dangerous in fact. In general, the accuracy of the prediction reached 86.67%, which verified that the dual-feedback Elman model was effective for situational prediction and worth further promotion and application in practice.

5 Conclusion

This study analyzed the prediction of network security situational level based on neural networks. The BPNN model was used for situational assessment, and the dual-feedback Elman model was used for situational prediction, and the network situational data were collected through the actual built network environment for experimental analysis. The results demonstrated that the accuracy of the BPNN model in predicting situations was 90%, and the dual-feedback Elman model had an accuracy of 86.67% in predicting the situational level. The two models have high stability and can effectively predict the network security situational level to achieve network security.

References

[1] Xhafa F, Li J, Kolici V (2015). SPECIAL ISSUE: Advances in Secure Data Streaming Systems. Informatica, 39, pp. 337-337.

[2] Zhang YN, Zhang MQ, Wang XA, Niu K, Liu J (2016). A Novel Video Steganography Algorithm Based on Trailing Coecients for H.264/AVC. Informatica, 40, pp. 63-70.

	Predicted results	Actual situation
1	Safe	Safe
2	Mildly dangerous	Mildly dangerous
3	Generally dangerous	Generally dangerous
4	Generally dangerous	Generally dangerous
5	Generally dangerous	Generally dangerous
6	Highly dangerous	Moderately dangerous
7	Safe	Safe
8	Generally dangerous	Generally dangerous
9	Generally dangerous	Generally dangerous
10	Safe	Safe
11	Mildly dangerous	Mildly dangerous
12	Generally dangerous	Generally dangerous
13	Generally dangerous	Generally dangerous
14	Moderately dangerous	Highly dangerous
15	Mildly dangerous	Mildly dangerous
16	Generally dangerous	Generally dangerous
17	Highly dangerous	Highly dangerous
18	Safe	Safe
19	Generally dangerous	Mildly dangerous
20	Generally dangerous	Generally dangerous
21	Generally dangerous	Generally dangerous
22	Mildly dangerous	Mildly dangerous
23	Generally dangerous	Generally dangerous
24	Moderately dangerous	Moderately dangerous
25	Safe	Mildly dangerous
26	Mildly dangerous	Mildly dangerous
27	Mildly dangerous	Mildly dangerous
28	Generally dangerous	Generally dangerous
29	Generally dangerous	Generally dangerous
30	Mildly dangerous	Mildly dangerous

Table 10: Comparative results of security situational levels.

[3] Ponnuramu V, Tamilselvan L (2016). Secured Storage for Dynamic Data in Cloud, Informatica, 40, pp. 53-61.

[4] Kayashima M, Terada M, Fujiyama T, Koizumi M, Katou E (2015). VPN construction method for multiple firewall environment. Systems & Computers in Japan, 31, pp. 57-63. https://doi.org/10.1002/1520-684X(200012)31:14<57::AID-SCJ7>3.0.CO;2-M

[5] Gogoi P, Borah B, Bhattacharyya DK (2013). Network Anomaly Identification using Supervised Classifier. Informatica, 37, pp. 93-105.

[6] Xu G, Cao Y, Ren Y, Li X, Feng Z (2017). Network Security Situation Awareness Based on Semantic Ontology and User-defined Rules for Internet of Things. IEEE Access, pp. 1-1. https://doi.org/10.1109/ACCESS.2017.2734681

[7] Zhao DM, Liu JX (2018). Study on Network Security Situation Awareness based on Particle Swarm Optimization Algorithm. Computers & Industrial Engineering, pp. S036083521830007X. https://doi.org/10.1016/j.cie.2018.01.006

[8] Zhang H, Yi Y, Wang J, Cao N, Duan Q (2018). Network Security Situation Awareness Framework based on Threat Intelligence. Computers, Materials & Continua, 56, pp. 381-399. https://doi.org/10.3970/cmc.2018.03787

[9] Li JZ, Meng XR, Wen XX, Kang QY (2015). Network security situation prediction based on Gaussian process optimized by glowworm swarm optimization. Systems Engineering & Electronics, 37, pp. 1887-1893. https://doi.org/10.3969/j.issn. 1001-506X.2015.08.26

[10] Hu J, Ma D, Liu C, Shi Z, Yan H, Hu C (2019). Network security situation prediction based on MR-SVM. IEEE Access, PP, pp. 1-1. https://doi.org/10.1109/ACCESS.2019.2939490

[11] Muehlethaler C M, Knecht C P (2016). Situation Awareness Training for General Aviation Pilots using Eye Tracking. IFAC Papersonline, 49, pp. 66-71. https://doi.org/10.1016/j.ifacol.2016.10.463

[12] Holsopple J, Yang S J, Sudit M (2015). Mission Impact Assessment for Cyber Warfare. Studies in Computational Intelligence, 563, pp. 239-266. https://doi.org/10.1007/978-3-319-08624-8_11

[13] Zhang H, Huang Q, Li F, Zhu J (2016). A network security situation prediction model based on wavelet neural network with optimized parameters. Digital Communications and Networks, pp. 139-144. https://doi.org/10.1016/j.dcan.2016.06.003

[14] Li F W, Zhang X Y, Zhu J, Huang Q (2016). Network security situation prediction based on APDE-RBF neural network. Systems Engineering and Electronics, 38, pp. 2869-2875. https://doi.org/ 10.3969/j.issn.1001-506X.2016.12.28

[15] Wang ZP (2010). Research on network security situation assessment based on index system. National University of Defense Technology.

[16] Liu S, Ren T, Mu H, Yan G (2016). Temperature prediction of the molten salt collector tube using BP neural network. IET Renewable Power Generation, 10, pp. 212-220. https://doi.org/10.1049/iet-rpg.2015.0065

[17] Liu W, Liu CK, Zhuang DM, Liu ZQ, Yuan XG. (2012). Study on Pilot Performance Model. Advanced Materials Research, 383-390, pp. 2545-2549. https://doi.org/10.4028/www.scientific.net/AMR.383-390.2545

[18] Liu H, Tian HQ, Liang XF, Li YF (2015). Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks. Applied Energy, 157, pp. 183-194. https://doi.org/10.1016/j.apenergy.2015.08.014