Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 69 (2014) 1209 – 1215

Procedia Engineering

www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

Application of Neural Networks in Computer Security

Halenar Igor*, Juhasova Bohuslava, Juhas Martin,Nesticky Martin

The Institute of Applied Informatics, Automation and Mathematics, Slovak University of Technology, Trnava, Slovakia

Abstract

This contribution is focused on the insurance of control system data communication via neural network technologies in connection with classical methods used in expert systems. The solution proposed defines a way of data element identification in transfer networks, solves the transformation of their parameters for neural network input and defines the type and architecture of a suitable neural network. This is supported by experiments with various architecture types and neural network activation functions and followed by subsequent real environment tests. A functional system proposal with possible practical application is the result.

Keywords: communication; neural network; security

1. Introduction

Production process controls are being replaced practically everywhere by discrete automated control systems. This dynamic development is rapidly increasing the importance of automation control systems and processes. The information transfer to the control system has to be as immediate as possible; a direct on-line connection is preferred. For properly working, the control system itself needs the information on results and feedback. All these parts necessarily require properly working communication via channels capable of transferring the essential information quickly and reliably.

* Corresponding author. Tel.: ++421 33 5511033; fax: ++421 906 068299
E-mail address: igor.halenar@stuba.sk

doi:10.1016/j.proeng.2014.03.111

2. Subject matter formulation

The aims can be summarized as follow. For the first analyze the security of control systems communication. For next we analyze possible neural networks utilization by the data transfer validation and select a suitable neural network type. Then we propose a functional security system model and using suitable tools, execute the model system of communication network security.

The control process is based on current information on the values of input and controlled quantities, on the state of individual system parts and other relevant information. Communication channels represented by the file of protocols as well as physical transfer media are used to transfer the values. In present day is mostly used a network communication based on TCP/IP protocols. The trends in production systems are to combine types of industrial ethernet networks and traditional technologies. The result of this is the creation of various additions to TCP/IP protocol, which are solving compatibility problems, but also bring to the automation network errors and security risks. The protection is happening on several levels and is performed by various systems such as firewalls, IDS (Intrusion Detection System), IPS (Intrusion Prevention System) devices or protocols.

2.1. System design

The proposal of the entire system consists of more parts. Individual steps of the functional model implementation to ensure the communication via neural networks correspond to the stages of the implementation. Necessary steps for practical implementation of the proposed system can be summarized as follow:

- Necessity (by some suitable way) to implement the main core of the security system
- Propose, carry out and implement the functional model of the neural network, meeting the requirements for possible implementation as proposed
- Select a suitable element of active security of the protected network, which is able to dynamically change the parameters of communication set data transfer on the basis of the neural network module’s outputs and implement the solution by suitable software

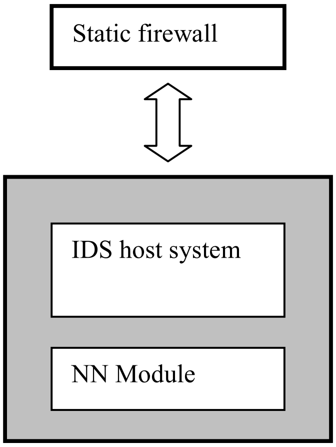

Fig. 1. System design.

The suggested system is able to adapt dynamically to the possible failure of a communication, system attack or more complex forms of infiltration (exploits, out of band communications, the analysis of covert communication channels).

The whole proposed system is suitable to build on existing IDS. Of course the existing system must meet certain requirements. For example there must be possibility to program own extension modules, possibility to capture packets and save captured packets. This is important point because of the possibility of repeating cycles of learning a neural network. As a host system is for us suitable IDS Snort (GNU GPL licence) [3].

2.2. Selection of a suitable neural network type

With the application and selection of a suitable neural network type, it is necessary to realize the implementation of any neural network has both advantages and disadvantages. The flexibility of neural network is an advantage - in addition, the neural network is able to analyze incomplete data. Non-linearity of data flows in communication networks is another aspect influencing the selection. Since the neural network output is expressed as probability, neural network outputs can subsequently work as a certain prediction. Because neural networks can improve their abilities by learning, the output information could then be used to generate various actions in cases where a prediction is an alert of an attack attempt.

There are many types of internal arrangements of neural networks, either with or without controlled learning. According to sources available and utilization required, we select a feed-forward neural network with learning with backward spread of misuse. This neural network type provides sufficient flexibility and applicability for a large scale of tasks, where NN technologies can be utilized.

In our case, we assume nine input neurons and two output neurons. Number of neurons in input layer is given by number of parameters for describing the communication. We are using neural network for classification to two groups. There can be only one neuron in output layer. But for software realization reasons we have to use two output neurons. Other parameters of neural network are next. The activation function of hidden layer was used in all cases the standard sigmoid activation function:

$$x_i = f(in_i) = \frac{1}{1+e^{-cin_i}} \tag{1}$$

The activation function of output layer in the various experiments was a linear activation function:

$$x_i = f(in_i) = in_i \tag{2}$$

2.3. Data capturing and transformation

For learning and testing of the neural network we need large sample of captured traffic. Data can be from real environment or generated.

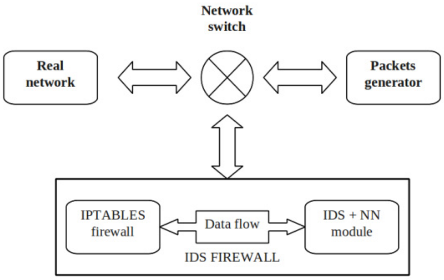

Fig. 2. The scheme for data recording.

For testing purpose in this case we are using some mixture - data from simulated network communication in laboratory environment and real traffic. After preliminary analysis of options and the available literature [1], [2], we propose that the characteristic of data communication in this case will be specified with following parameters:

- ID protocol - the protocol type associated with packet
- Source port - Number of TCP / UDP port of the source system
- Destination port - Number of TCP / UDP port of the target system
- Source Address - IP address of the source system
- Destination Address - IP address of the target system
- ICMP type - the type of ICMP packet
- Length of data transferred - the size of the data packet in bytes
- FLAGS - signs in the protocol header
- TCP window size - window size parameter of TCP packet

These data must be properly transformed as input to neural network. According to available sources [4] we can choose use of feed-forward neural network learning method with back-error propagation. This type of neural network provides sufficient flexibility and applicability to a wide range of tasks, where it is possible to use technology to minimize the objective function NN. To minimize the objective function we can use several optimization methods commonly used to minimize in the numerical mathematics [4].

The commonly used methods are including gradient methods, which disadvantage is the high number of iteration steps. Because this disadvantage we can use a variety of other, more efficient and faster optimization methods for adaptation of neural networks. In the available literature these methods are represented by the name "quickprop" (based on Newton's method) or other numeric methods (method of variable steps, entropic normalization models, least squares, etc.) . Alternatively, you can use modern methods to minimize using genetic algorithms or some of data analyzing methods as described in other literature [5], [6], [7]. The whole process of data transformation is necessary to perform simply and fast. For this, we have to implement programs in some text processing language. For example in this case we are using AWK executable script for Bourne Shell system. Adaptations must be carried out in several stages. The first step is extraction of the required values of the entire data package. We can afford to ignore the account records of the communications network service (ARP, RARP communication) and data, to us in terms of work, unattractive. The next phase is an appropriate representation of some elements. Specifically, it is the following parameters: ID protocol, FLAG and ICMP type.

For example in table1 is shown FLAG parameter transformation.

Table 1. Parameter FLAG - transformation.

Parameter FLAG	Substitution
NoFlag	1
RST	2
FIN	3
PSH	4
URG	5
SYN	6
OTHER	7

The transformation process should be fast and automatic. After processing by the AWK program we receive data represented by the following example in table2.

Table 2. Neural network input example.

Pattern number	Neural network input
1	1,1471751309,0,147175134254,0,0,0,44,1
2	6,62240183148,80,147175134254,2190,8514,4,1500,0
3	17,19416092,33239,233104778,1234,0,0,1500,0
4	6,147175130212,2246,19512215120,80,65535,1,40,0
5	17,13015680151,4262,2242127254,9875,0,0,224,0
6	1,147175130212,0,10254247189,0,0,0,36,2
…	…
…	…

Comma separated fields from first are IDPRO, SIP (source ip), SPO (source port), DIP (destination_ip), DEP (destination port), TW (size of tcp window), FLG (flag), SZ (size of packet), ICMPT (icmp type).

Given the expected use of neural networks (classification) is sufficient the distribution of patterns into two groups. This means to divide the patterns to correct packets (CLASS = 1) and those which will be subject to further testing (CLASS = 2). Designation of records is necessary for training neural networks. Records of class packets are stored in a separate table.

3. Practical experiments

With data transformation was selected 5000 models of the ongoing communication in the test environment. Of this number, we have developed a selective choice of four groups containing 200 models for learning neural network and a group of 100, 200 and 3000 models for testing. In experimental phases were made several attempts and tests with different number of neurons in the hidden layer. Specifically, the network involved in the 9-6-2, 9-10-2, 9-20-2 and 9-40-2 (input - hidden - output layer). As the algorithm to minimize the objective function was chosen quickprop method based on Newton's method [4].

3.1. Experiment 1 - six neurons in hidden layer

In the first experiment (nn 9-6-2) the absolute error of neural network after one thousand iterations was 0,17905. This value is stable and is not changing after 200 training cycles.

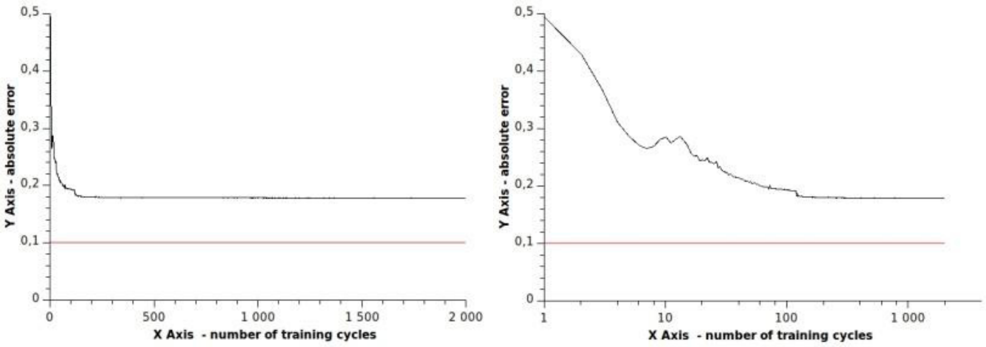

Fig. 3. Outputs from experiment 1 shown in linear and logarithmic scale.

3.2. Experiment 2 - ten neurons in hidden layer

The output from the neural networks with 10 neurons in hidden layer is similar to experiment one. The best value of the absolute error achieved in this experiment is 0,16937.

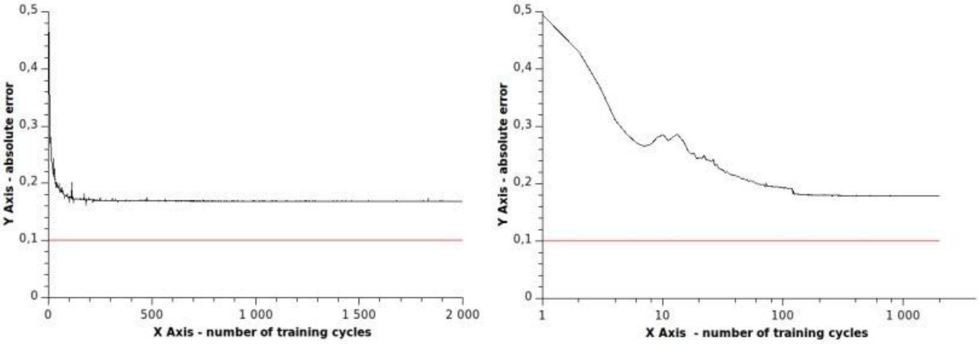

Fig. 4. Output from experiment 2 shown in linear and logarithmic scale.

3.3. Experiment 3 - twenty neurons in hidden layer

In the third case, the result obtained after one thousand iterations correspond to the absolute error is 0.06041.

The number of patterns in training set, which have achieved the required tolerance, was increased to 96.67%.

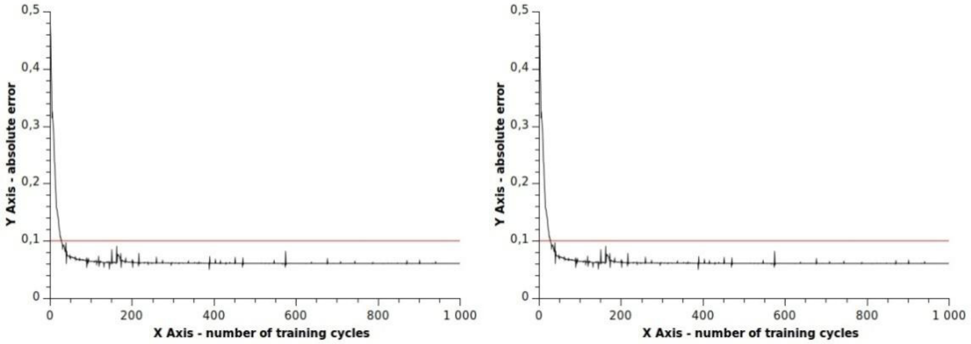

Fig. 5. Outputs from experiment 3 shown in linear and logarithmic scale.

3.4. Experiment 4 – forty neurons in hidden layer

In the fourth experiment was used neural network with 40 neurons in hidden layer. Such as in other experiments, the total number of specimens in the test group was one hundred. In this case we are getting best values from all. The number of samples in the test group which is satisfying the tolerance (0.10) is 98.56%.

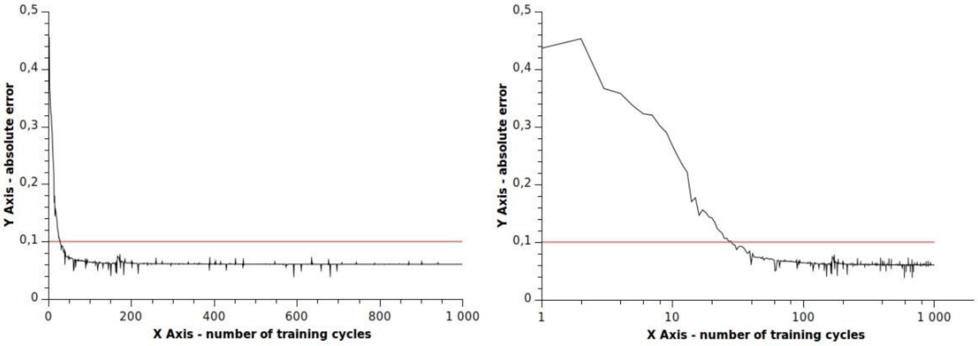

Fig. 6. Outputs from experiment 4 shown in linear and logarithmic scale.

4. Conclusion

The red line in graphs represents the absolute error tolerance, which is in our case 0,1. From the above results is clear, that given tolerance (abs. error less than 0,1) meets neural network with 20 and 40 neurons in hidden layer. But the best classification ability of the samples reaches neural network with 40 neurons in the hidden layer. Other experiments with more types of neural networks have shown that more neurons in hidden layer (50, 60 and 100) have no significant effect.

We can see in those experiments that designed and trained neural network is able to sufficiently classify individual data packets. Also we can say, that higher number of neurons in hidden layer has impact on sorting quality of neural network, but the in this case seems that number of 40 neurons in hidden layer is the best.

This area is very complex and there is still a lot of work to do. This work represents just a part of the overall problem. In the next we can solve number of problems. For example propose a neural network, much larger, which would be able to separate incoming packets to more classes. Than we can to extend the model with other neural networks that would search in the data stream for other anomalies [4]. Probably we can modify this model to use other type of neural network, for example some type of the Kohonen network.

References

[1] M. Thottan, Ch. Ji, “Anomaly Detection in IP Networks” Journal of Network and Systems Management : IEEE Transactions on signal processing, vol. 51, no. 8. New York: Springer, 2007 pp. 267 – 283.

[2] G. Conti, A. Kulsoom, “Passive visual fingerprinting of network attack tools”, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security table of contents. Washington DC, USA: ACM, Washington DC, 2004, pp. 45-54.

[3] M. Roesch, Ch. Greem (2009), “SNORT Users manual 2.8.5, The Snort Project”. [Online]. Available: http://www.snort.org/docs

[4] P. Sinčák, G. Andrejková, “Neurónové siete - Inžiniersky prístup (1. diel)”, (Neural networks - Master Access, part 1) Košice, Slovakia, Technická Univerzita Košice, 2005.

[5] R. Halenar “Contribution of Near Real Time ETL” 2011 International Conference on Database and Data Mining (ICDDM 2011) .Proceedings / editors: Steve Thatcher and Liu Guiping. - Sanya : IEEE, 2011. ISBN 978-1-4244-9610-5. pp. 243-24.

[6] M. Kebísek and P. Schreiber “The possibility of utilization of neural networks at the data mining”, CO-MAT-TECH 2004 : International Scientific Conference. Trnava, Slovak Republic, 14-15 October 2004. ISBN 80-227-2117-4. - pp. 589-595.

[7] CHing-Long Su, S.M. Yang, W.L. Huang “A two-stage algorithm integrating genetic algorithm and modified Newton method for neural network training in engineering systems” Expert Systems with Applications. ISSN: 0957-4174,15 Spetember 2011, pp. 12189 – 12194.