ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту. Вісник Дніпропетровського національного університету залізничного транспорту, 2018, № 2 (74)

МОДЕЛЮВАННЯ ЗАДАЧ ТРАНСПОРТУ ТА ЕКОНОМІКИ

UDC [004.7-047.72]:656.2

I. V. ZHUKOVYTS'KYY, V. M. PAKHOMOVA[2*]

[1*]Dep. «Electronic Computing Machines», Dnipropetrovsk National University of Railway Transport named after Academician V. Lazaryan, Lazaryan St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail ivzhuk@ua.fm, ORCID 0000-0002-3491-5976

[2*]Dep. «Electronic Computing Machines», Dnipropetrovsk National University of Railway Transport named after Academician V. Lazaryan, Lazaryan St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail viknikpakh@gmail.com, ORCID 0000-0002-0022-099X

IDENTIFYING THREATS IN COMPUTER NETWORK BASED ON MULTILAYER NEURAL NETWORK

Purpose. Currently, there appear more often the reports of penetration into computer networks and attacks on the Web-server. Attacks are divided into the following categories: DoS, U2R, R2L, Probe. The purpose of the article is to identify threats in a computer network based on network traffic parameters using neural network technology, which will protect the server. **Methodology.** The detection of such threats as Back, Buffer_overflow, Quess_password, Ipsweep, Neptune in the computer network is implemented on the basis of analysis and processing of data on the parameters of network connections that use the TCP/IP protocol stack using the 19-1-25-5 neural network configuration in the Fann Explorer program. When simulating the operation of the neural network, a training (430 examples), a testing (200 examples) and a control sample (25 examples) were used, based on an open KDDCUP-99 database of 500000 connection records. **Findings.** The neural network created on the control sample determined an error of 0.322. It is determined that the configuration network 19-1-25-5 copes well with such attacks as Back, Buffer_overflow and Ipsweep. To detect the attacks of Quess_password and Neptune, the task of 19 network traffic parameters is not enough. **Originality.** We obtained dependencies of the neural network training time (number of epochs) on the number of neurons in the hidden layer (from 10 to 55) and the number of hidden layers (from 1 to 4). When the number of neurons in the hidden layer increases, the neural network by Batch algorithm is trained almost three times faster than the neural network by Resilient algorithm. When the number of hidden layers increases, the neural network by Resilient algorithm is trained almost twice as fast as that by Incremental algorithm. **Practical value**. Based on the network traffic parameters, the use of 19-1-25-5 configuration neural network will allow to detect in real time the computer network threats Back, Buffer_overflow, Quess_password, Ipsweep, Neptune and to perform appropriate monitoring.

Keywords: network traffic; threat; neural network; sampling; hidden layer; hidden neurons; training algorithm; number of epoch; error

Introduction

There have recently been increasingly frequent computer penetration reports and attacks on Web-server. Very often, intruders bypass established protective devices. Attacks are carried out in a very short time and the variety of threats is constantly increasing, which prevents from detecting and preventing them with standard protective equipment [2–3]. Existing approaches are characterized by a number of features that hinder their use: low speed of work; poor accuracy [1, 13, 15]. To eliminate these shortcomings, a neural network technology is proposed [1–4, 6–12]: multilayer perceptron; Kohonen network; neuronetty network (hybrid system).

Attacks are divided into the following categories [2, 14–15]: DoS (Back, Land, Neptune, Pod,

doi 10.15802/stp2018/130797

ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту. Вісник Дніпропетровського національного університету залізничного транспорту, 2018, № 2 (74)

Smurf, Teardrop), U2R (Buffer_overflow, Loadmodule, Perl, Rootkit), R2L (Ftp_write, Quess_passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster), Probe (Ipsweep, Hmap, Portsweep, Satan). DoS attack is characterized by the generation of a large amount of traffic, which leads to overload and lockup of the server; U2R attack (User to Root) involves receiving by a registered user of administrator's privileges; R2L attacks (Remote to Local) are characterized by access of an unregistered user to a computer from a remote machine; Probe attacks include scanning of ports in order to obtain confidential information.

At the present stage, various solutions are offered for the modernization of existing computer networks, in particular, in the information and telecommunication system of the Prydniprovska railway [5, 16].

Purpose

To develop a method for detecting threats in a computer network based on network traffic parameters using a multi-layer neural network in the Fann Explorer program.

Methodology

To detect threats we used 19 network traffic parameters (x_i) [14]: package type (TCP, UDP, ICMP and others); service type (http, telnet, ftp_data, eco_i, private); flag; number of bytes from source to recipient; number of bytes from recipient to source; number of hot indicators; successful entrance; number of compromised conditions; number of connections to the host for 2 s; number of connections to one service for 2 s; percentage of connections to the service for 2 s; percentage of connection to different hosts; number of connections to the local host installed by the remote side for 2 s; number of connections to the local host installed by the remote side, uses 1 service; percentage of connections to this service; percentage of connections to other services; percentage of connections to the host with the port number of the source; percentage of connections with a rej-type error for the recipient host; percentage of connections with a rej-type error for the service. On the basis of the values of the set of the attack features, it is necessary to carry out the classification conclusion (y_1, y_2, y_3, y_4, y_5) of the following threats: Back, Buffer_overflow, Quess_password, Ipsweep, Neptune. Detection of threats in a computer network is based on the analysis and processing of data on the parameters of network connections using the TCP/IP protocol stack. As initial data we used KDDCUP-99 database of 5,000,000 connection records (the sequence of TCP packets for the final period, the start and end points of which are clearly defined, during which the data is transmitted from the sender's IP address to the recipient's IP address using the defined protocol). As a mathematical tool of problem solution we took the neural network (NN) of the configuration 19-1-25-5, where 19 is the number of neurons in the input layer, 1 is the number of hidden layers, 25 is the number of neurons in the hidden layer, and 5 is the number of neurons in the resulting layer.

Findings

Purpose and features of the Fann Explorer program. Fann Explorer is portable graphical environment for developing, training and testing neural networks that supports animation training, creation, provides an easy to use browser-based interface for fast artificial neural network (Fann library). FannKernel provides a neural network with the kernel, which is a multi-threaded kernel, so several neural networks can be studied and explored at the same time. This program has a fairly wide-ranging functionality for training and researching neural networks. The View menu allows you to open three main windows: Controller (main panel, which defines all the parameters for teaching the neural network), Error plot (dependence graph of the mean square error on the number of passed training epochs), Weight Graph. On the Topology tab, you can get acquainted with the main characteristics of the topology (Fig. 1).

The Testing menu allows you to check how well the neural network is being trained and the ability to edit incoming data. The test results are presented as a comparison graph of the mean square error with the reference value shown in Fig. 2.

doi 10.15802/stp2018/130797

ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту. Вісник Дніпропетровського національного університету залізничного транспорту, 2018, № 2 (74)

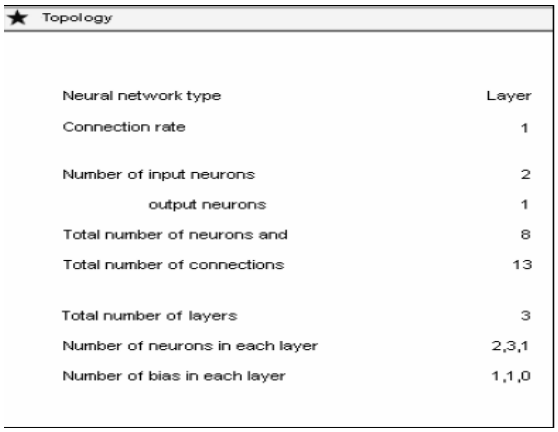

Fig. 1. Description of network topology

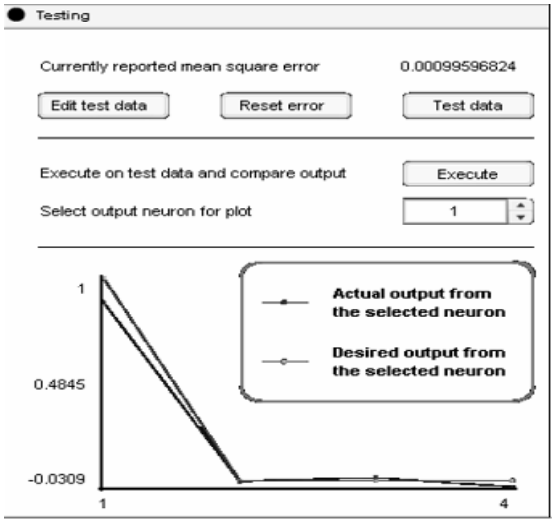

Fig. 2. Testing menu

Preparation of samples (preparatory stage). The following samples were prepared: training (430 examples), test (200 examples), control (25 examples). Since NN does not accept textual information, samples need to be processed by marking the text parameters with numerical values. The changes apply to the first three columns of the sample and the attack tags, namely: package type (tcp – 1; icmp – 2); type of service (http – 1; telnet – 2; ftp_data – 3; ftp – 4; eco_i – 5; private – 6; smtp-7; nhsp-8; http_443-9); flag (SF-1; RSTR-2; RSTO-3; S0-4; S3-5); Attack tag (Back – 0 0 0 0 1; Buffer_overflow – 0 0 0 1 0; Quess_pwd – 0 0 1 0 0; Ipsweep – 0 1 0 0 0; Neptune – 1 0 0 0 0). Sampling is given in any text editor (all values are separated by spaces); file has extension .train. A fragment of the processed training sample is shown in Fig. 3

Fig. 3. Fragment of the processed training sample

Creating neural network in Fann Explorer. Fann Explorer (Fann Artificial Neural Networks) from Macromedia Inc is a portable environment for the development, training and testing of neural networks; supports animation of the training process; implements multilayer artificial neural networks; has a multi-threaded kernel that provides computing. The Neural Network was created with the participation of the student Mamenko D. V., setting of NN architecture is shown in Fig. 4

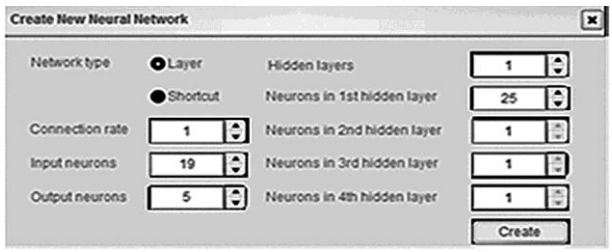

Fig. 4. Setting of NN architecture

Training of neural network. The purpose of NN training is to match such values of its parameters, in which the error of training is minimal. In the Fann Explorer program, the network training options are set using the Train tab (Training algorithm).

Incremental algorithm is a standard reverse error propagation algorithm where weighting factors are updated after each training period. This means that weights are updated many times during one training.

Resilent algorithm is a batch training algorithm that provides good results for many tasks, the algorithm is adaptive and does not use Training rate for training.

Quick algorithm uses the Training rate parameter and gives good results when solving problems.

Batch method is a standard reverse propagation algorithm where the weights are updated after calculating the mean square error for the entire training sample. Since the average square of the error is calculated more correctly than in the sequential

doi 10.15802/stp2018/130797

ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту. Вісник Дніпропетровського національного університету залізничного транспорту, 2018, № 2 (74)

algorithm, some tasks will achieve more effective solutions with this algorithm.

Error Function: Linear is a linear error function that calculates it as the difference between the actual result given by the neural network and the expected value that the operator has set; Tangential (Tanh) is a function of error, which makes a large deviation during training. The idea of the function is that it is better if 10 neurons have an error of 10% at the output, than one of them will have an error of 100%. This function is the default error, but it can lead to poor training outcomes if you set the Training rate too high.

The functions of Hidden layer activation and Output layer activation are as follows: Symmetric, Asymmetric-Linear, Sigmoid, Stepwize, Threshold.

The Training menu manages the training process of NN, sets the adaptation parameters of the NN link weight factors such as: the number of training periods, the value of the mean square error, the initial values of the NN link weight factors using the Initialize button.

Testing of neural network. Figure 5 shows a graph of NN testing of 19-1-25-5 configuration after training and testing on the appropriate samples. The bright line on the graph shows the expected response, while the darker one is the actual response. The test error is 0.05868; the lines almost coincide, so NN is well suited to the task.

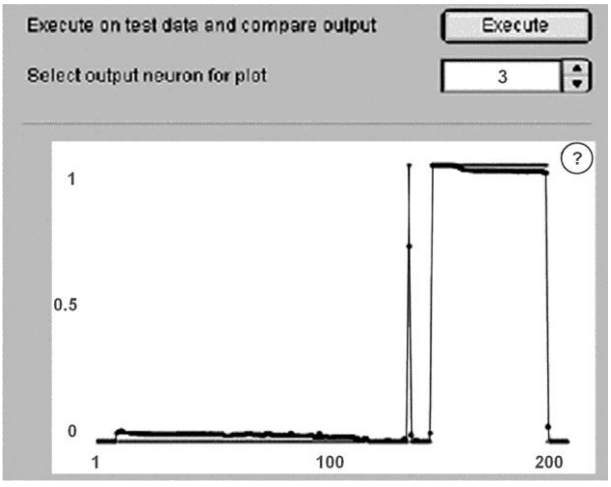

Fig. 5. NN testing schedule

Analysis of the results. The NN of 19-1-25-5 configuration was trained according to the standard algorithm for error propagation and the training speed of 0.699. Activation function is sigmoid (symmetric); error function is tangential. NN passed 6,000 training epochs. After training (430 examples) and testing (200 examples), NN determined the error of 0.1066, as shown in Fig. 6

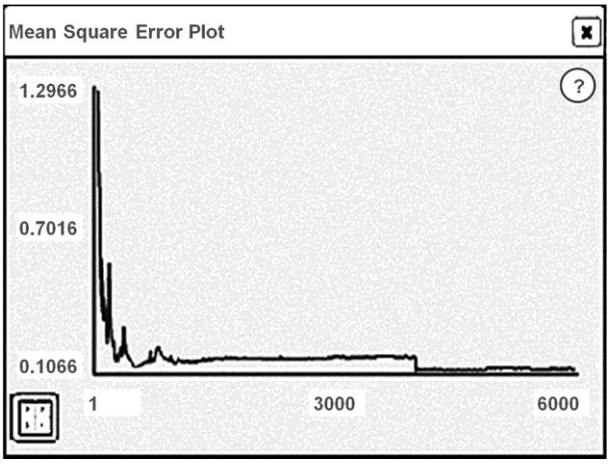

Fig. 6. Error after NN training

But when testing on a control sample, which consisted of 25 examples for each of the five threats, NM determined an error of 0.322. Thus, the NN copes well with Back, Buffer_overflow and Ipsweep attacks, but it does not recognize the Quess_password and Neptune attacks. From the test schedule on the control sample, it can be seen that the first neuron responsible for detecting the Back attack detected four of the five threats (the darker line is an expected solution, and the lighter is the actual one), Fig. 7 (a).

Also, NN recognized all five threats of Buffer_overflow type, fig. 7 (b). The lines on the chart are almost the same, but also one of the threats of Back type is incorrectly assigned by NN to the Buffer_overflow class. The obtained results of the experimental study are presented in Table 1.

Study of network training time versus the number of hidden neurons. The study was carried out on NN with different number of neurons in a hidden layer: from 10 to 55. Experiments were carried out on the following models: Model No. 1 (Initialize Resilent Sigmoid Stepwise Algorithm), Model No. 2 (Randomize Batch Sigmoid Symmetric Algorithm). The results of studies are presented in Table 2.

doi 10.15802/stp2018/130797

ISSN 2307–3489 (Print), ISSN 2307–6666 (Online)

Наука та прогрес транспорту. Вісник Дніпропетровського національного університету залізничного транспорту, 2018, № 2 (74)

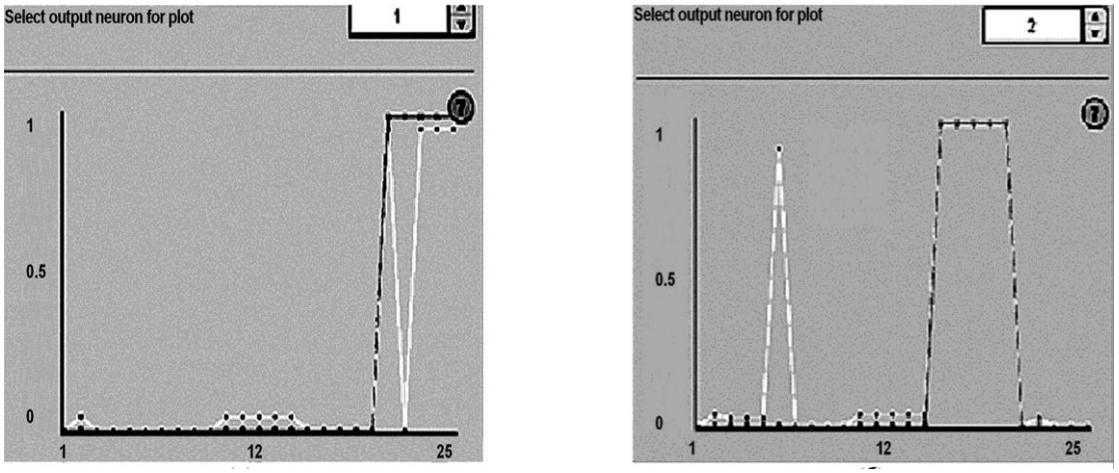

Fig. 7. Graph of the resulting neuron of the network:
a – the first one; *b* – the second one

Table 1

Results of NN operation on a control sample

Attack	Number of attacks	Number of detected attacks	Number of recognition errors
Back	5	4	1
Buffer_overflow	5	5	1
Quess_password	5	2	5
Ipsweep	5	4	1
Neptune	5	1	5

Table 2

Dependence of NN training time on the number of hidden neurons

Number of neurons in the hidden layer	Model No. 1		Model No. 2	
	Error	Number of training epochs	Error	Number of training epochs
10	0.1982	20 039	0.1996	6 387
15	0.1994	253	0.2021	2 000
25	0.197	214	0.1993	33
35	0.196	170	0.1977	22
45	0.1964	116	0.1960	18
55	0.199	249	0.193	35

When increasing the number of neurons in the hidden layer, the NN is trained faster, but in some cases, when the optimal amount starts to exceed (> 45), the training rate falls. In addition, model No. 2 is trained much faster than model No. 1. The constructed graphs of dependence of the number of epochs on the number of neurons in the hidden layer under different training algorithms are presented in Fig. 8.

doi 10.15802/stp2018/130797

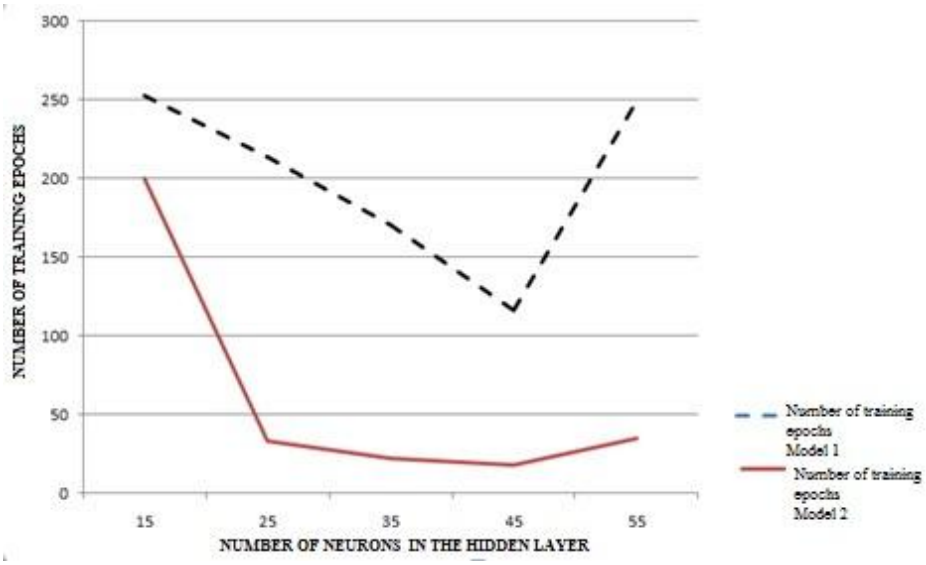

Fig. 8. Dependence of the number of epochs on the number of hidden neurons

Study of training time versus the number of hidden layers. The study was carried out on NN with a different number of hidden layers: from 1 to 4; 15 formal neurons in each. Experiments were carried out on the following models: Model No. 3 (Incremental Training rate 0.4), Model No. 4 (Resilient Training rate 0.8). The initialization algorithm was Randomize, activation function – sigmoid (symmetric). The results of experimental studies are listed in Table. 3.

Table 3

Dependence of NN training time on the number of hidden layers

Number of hidden layer	Model No. 3		Model No. 4	
	Error	Number of training epochs	Error	Number of training epochs
1	0.1495	93 775	0.1694	164
2	0.1994	551	0.01394	178
3	0.1279	484	0.1413	224
4	0.1435	2272	0.1478	547

The graphs of the dependence of the number of training epochs on the number of hidden layers based on different training algorithms are plotted and shown in Fig. 9.

From the figure it can be seen that when increasing the number of hidden layers (> 3) in NN, training accelerates only to a certain point, until the layers become too many, then the network begins to slow down. In addition, the model No. 4 is trained almost twice as fast as the model No. 3, but has a bigger error, which results from an increased training rate.

Originality and practical value

The originality lies in the fact that there are found dependencies of the training time (number of epochs) of the multilayer neural network on the number of hidden layers and hidden neurons according to different training algorithms. The practical value is that the network traffic parameters, using the 19-1-25-5 configuration neural network, will allow in real-time to detect the threats of Back, Buffer_overflow, Quess_password, Ipsweep, Neptune on the computer network and carry out appropriate control.

doi 10.15802/stp2018/130797

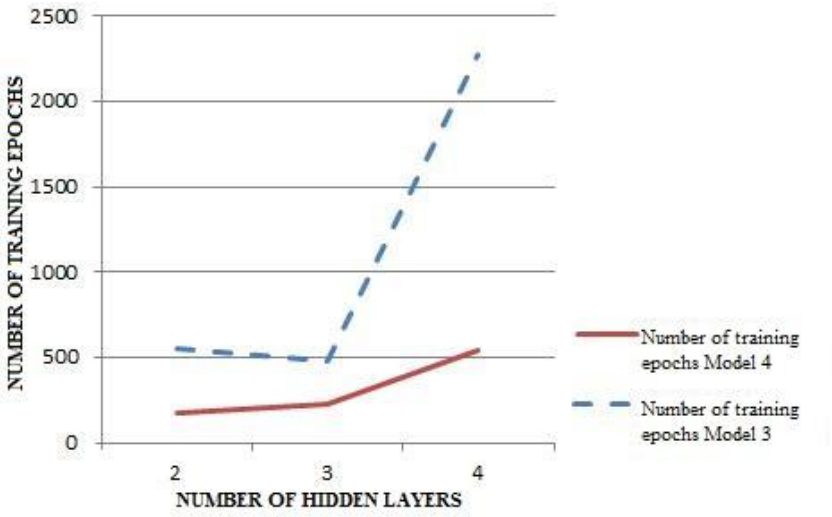

Fig. 9. Dependence of the number of training epochs on the number of hidden layers

Conclusions

1. At the preparatory stage, based on the data of the KDDCUP-99 database, the following samples were generated: training (430 examples), test (200 examples), control (25 examples).

2. To determine the attacks (Back, Buffer_overflow, Quess_password, Ipsweep, Neptune) in the computer network, the Fann Explorer program created NN of 19-1-25-5 configuration, with 19 network traffic parameters inputted; training, testing (error 0.1) and evaluation of the received results (error 0.3) on the corresponding samples were conducted. In particular, the first neuron responsible for recognizing the Back attack detected four of the five threats.

3. Experimental studies of the dependence of the training time (number of epochs) on the number of hidden neurons (from 10 to 55) on NN were conducted: Model No. 1 (Resilient algorithm), Model No. 2 (Batch algorithm); the Batch algorithm NM is trained three times faster than the NN based on the Resilient algorithm. The experimental studies of the dependence of the training time on the number of hidden layers (from 1 to 4) on the NN were conducted: Model No. 3 (Incremental algorithm), Model No. 4 (Resilient algorithm); the Resilient algorithm NM is trained almost twice as fast as NN by the Incremental algorithm.

LIST OF REFERENCE LINKS

1. Гришин, А. В. Нейросетевые технологии в задачах обнаружения компьютерных атак / А. В. Гришин // Информационные технологии и вычислительные системы. – 2011. – № 1. – С. 53–64.
2. Жульков, Е. В. Построение модульных нейронных сетей для обнаружения классов сетевых атак [Electronic resource] : автореф. дис. … канд. техн. наук : 05.13.19 / Жульков Евгений Владимирович ; Санкт-Петербург. гос. политехн. ун-т. – Санкт-Петербург, 2007. – 15 с. – Available at: http://elib.spbstu.ru/dl/1501.pdf/view. – Title from the screen. – Accessed : 15.02.2018.
3. Корпань, Я. В. Комплекс методів та засобів захисту інформації в комп'ютерних системах / Я. В. Корпань // Мир науки и инноваций. – 2015. – Т. 3. – С. 31–35.
4. Марченко, А. К. Обнаружение атак в системах нейросетевыми средствами / А. А. Марченко, С. В. Матвиенко, Ф. Г. Нестерук // Науч.-техн. вестн. информационных технологий, механики и оптики. – 2007. – № 39. – С. 83–93.
5. Пахомова, В. М. Можливості модернізації комп'ютерної мережі інформаційно-телекомунікаційної системи Придніпровської залізниці / В. М. Пахомова // Інформ.-керуючі системи на залізн. трансп. – 2015. – № 5. – С. 32–38.

doi 10.15802/stp2018/130797

6. Пилюгина, К. Н. Применение нейронных сетей с целью обнаружения вторжений [Electronic resource] / К. Н. Пилюгина // Современные научные исследования и инновации. – 2016. – № 2. – Available at: http://web.snauka.ru/issues/2016/02/63248. – Title from the screen. – Accessed : 19.02.2018.
7. Писаренко, И. Нейросетевые технологии в безопасности [Electronic resource] / И. Писаренко // Information Security. – 2009. – № 4. – Available at: http://www.itsec.ru/articles2/Oborandteh/neyrosetevye-tehnologii-v-biznese. – Title from the screen. – Accessed : 19.02.2018.
8. Постарнак, Д. В. Критический анализ моделей нейронных сетей / Д. В. Постарнак // Вестн. Тюмен. гос. ун-та. Физико-математ. науки. Информатика. – 2012. – № 4. – С. 162–167.
9. Amini, M. Effective Intrusion Detection with a Neural Network Ensemble using Fuzzy Clustering and Stacking Combination Method / M. Amini, J. Rezaeenour, E. Hadavandi // Journal of Computing and Security. – 2015. – Vol. 1. – Iss. 4. – P. 293–305.
10. Amini, M. A Neural Network Ensemble Classifier for Effective Intrusion Detection using Fuzzy Clustering and Radial Basis Function Networks / M. Amini, J. Rezaeenour, E. Hadavandi // Intern. Journal on Artificial Intelligence Tools. – 2016. – Vol. 25. – Iss. 02. – P. 1550033. doi: 10.1142/s0218213015500335.
11. A Survey of Artificial Immune System Based Intrusion Detection / Hua Yang, Tao Li, Xinlei Hu, Feng Wang, Yang Zou // The Scientific World Journal. – 2014. – Vol. 2014. – P. 1–11. doi: 10.1155/2014/156790.
12. Branitskiy, A. Network attack detection based on combination of neural, immune and neuro-fuzzy classifiers / A. Branitskiy, I. Kotenko // 2015 IEEE 18th Intern. Conf. on Computational Science and Engineering : Conf. Paper (21–23 Oct. 2015). – Porto, Portugal, 2015. – P. 152–159. doi: 10.1109/cse.2015.26.
13. Cannady, J. Artificial Neural Networks for Misuse Detection / J. Cannady // Proc. of the 21st National Information Systems Security Conference (October 5–8, 1998). – Arlington, Virginia, 1998. – P. 443–456.
14. KDDCup1999Data [Electronic resource]. – Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. – Title from the screen. – Accessed : 19.02.2018.
15. Moradi, M. System for intrusion Detection and Classification of Attacks [Electronic resource] // Інформаційний портал університету Квінс. – 2013. – Available at: http://research.cs.queensu.ca/moradi/148-04-mm-mz.pdf. – Title from the screen. – Accessed : 19.02.2018.
16. Pakhomova, V. M. Network Traffic Forcasting in information-telecommunication System of Prydniprovsk Railways Based on Neuro-fuzzy Network / V. M. Pakhomova // Наука та прогрес транспорту. – 2016. – № 6 (66). – С. 105–114. doi: 10.15802/stp2016/90485.

І. В. ЖУКОВИЦЬКИЙ[1*], В. М. ПАХОМОВА[2*]

[1*]Каф. «Електронні обчислювальні машини», Дніпропетровський національний університет залізничного транспорту імені академіка В. Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (056) 373 15 89, ел. пошта ivzhuk@ua.fm, ORCID 0000-0002-3491-5976
[2*]Каф. «Електронні обчислювальні машини», Дніпропетровський національний університет залізничного транспорту імені академіка В. Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (056) 373 15 89, ел. пошта viknikpakh@gmail.com, ORCID 0000-0002-0022-099X

ВИЯВЛЕННЯ ЗАГРОЗ У КОМП'ЮТЕРНІЙ МЕРЕЖІ НА ОСНОВІ БАГАТОШАРОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Мета. Останнім часом все частіше з'являються повідомлення про проникнення в комп'ютерні мережі та атаки на Web-сервери. Атаки поділяють на наступні категорії: DoS, U2R, R2L, Probe. Метою статті є виявлення загроз у комп'ютерній мережі на основі параметрів мережного трафіка з використанням нейромережної технології, що дозволить захистити сервер. **Методика**. Виявлення в комп'ютерній мережі таких загроз, як Back, Buffer_overflow, Quess_password, Ipsweep, Neptune здійснено на основі аналізу та обробки даних про параметри мережних з'єднань, що використовують стек протоколів TCP/IP, із застосуванням нейронної мережі конфігурації 19-1-25-5 у програмі Fann Explorer. Під час моделювання роботи нейронної мережі використані навчальна (430 прикладів), тестова (200 прикладів) та контрольна (25 прикладів) вибірки, що складені на основі відкритої бази даних KDDCUP-99 із 5 000 000 записів про з'єднання. **Результати**. Створена нейронна мережа на контрольній вибірці визначила похибку в 0,322. Визначено, що мережа конфігурації 19-1-25-5 добре справляється з такими атаками, як Back, Buffer_overflow та Ipsweep. Для розпізнання атак Quess_password і Neptune недостатньо завдання 19 параметрів мережного трафіку. **Наукова новизна.** Отримані залежності часу навчання (кількості епох) нейронної мережі від кількості нейронів у прихованому шарі (від 10 до 55) та кількості прихованих шарів (від 1 до 4). За умови збільшення

doi 10.15802/stp2018/130797

кількості нейронів у прихованому шарі нейронна мережа за алгоритмом Batch навчається швидше майже в три рази, ніж нейронна мережа за алгоритмом Resilient. Якщо збільшити кількість прихованих шарів, нейронна мережа за алгоритмом Resilient навчається майже в два рази швидше, ніж за алгоритмом Incremental. **Практична значимість.** На основі параметрів мережного трафіка використання нейронної мережі конфігурації 19-1-25-5 дозволить у реальному часі виявити загрози Back, Buffer_overflow, Quess_password, Ipsweep, Neptune на комп'ютерну мережу та здійснити відповідний контроль.

Ключові слова: мережний трафік; загроза; нейронна мережа; вибірка; прихований шар; приховані нейрони; алгоритм навчання; кількість епох; похибка

И. В. ЖУКОВИЦКИЙ[1*], В. Н. ПАХОМОВА[2*]

[1*]Каф. «Электронные вычислительные машины», Днепропетровский национальный университет железнодорожного транспорта имени академика В. Лазаряна, ул. Лазаряна, 2, Днипро, Украина, 49010, тел. +38 (056) 373 15 89, эл. почта ivzhuk@ua.fm, ORCID 0000-0002-3491-5976

[2*]Каф. «Электронные вычислительные машины», Днепропетровский национальный университет железнодорожного транспорта имени академика В. Лазаряна, ул. Лазаряна, 2, Днипро, Украина, 49010, тел. +38 (056) 373 15 89, эл. почта ivzhuk.@ua.fm, ORCID 0000-0002-0022-099X

ВЫЯВЛЕНИЕ УГРОЗ В КОМПЬЮТЕРНОЙ СЕТИ НА ОСНОВЕ МНОГОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ

Цель. В последнее время все чаще появляются сообщения о проникновении в компьютерные сети и атаки на Web-серверы. Атаки подразделяют на следующие категории: DoS, U2R, R2L, Probe. Целью статьи является выявление угроз в компьютерной сети на основе параметров сетевого трафика с использованием нейросетевой технологии, что позволит защитить сервер. **Методика.** Обнаружение в компьютерной сети таких угроз как Back, Buffer_overflow, Quess_password, Ipsweep, Neptune осуществлено на основе анализа и обработки данных о параметрах сетевых соединений, что используют стек протоколов TCP/IP, с применением нейронной сети конфигурации 19-1-25-5 в программе Fann Explorer. При моделировании работы нейронной сети использованы учебная (430 примеров), тестовая (200 примеров) и контрольная (25 примеров) выборки, составленные на основе открытой базы данных KDDCUP-99 с 5 000 000 записей о соединении. **Результаты.** Созданная нейронная сеть на контрольной выборке определила погрешность в 0,322. Определено, что сеть конфигурации 19-1-25-5 хорошо справляется с такими атаками как Back, Buffer_overflow и Ipsweep. Для распознавания атак Quess_password и Neptune недостаточно задания 19 параметров сетевого трафика. **Научная новизна.** Получены зависимости времени обучения (количества эпох) нейронной сети от количества нейронов в скрытом слое (от 10 до 55) и количества скрытых слоев (от 1 до 4). При условии увеличения количества нейронов в скрытом слое нейронная сеть по алгоритму Batch учится быстрее почти в три раза, чем нейронная сеть по алгоритму Resilient. Если увеличить количество скрытых слоев, нейронная сеть по алгоритму Resilient учится почти в два раза быстрее, чем по алгоритму Incremental. **Практическая значимость.** На основе параметров сетевого трафика использование нейронной сети конфигурации 19-1-25-5 позволит в реальном времени обнаружить угрозы Back, Buffer_overflow, Quess_password, Ipsweep, Neptune на компьютерную сеть и осуществить соответствующий контроль.

Ключевые слова: сетевой трафик; угроза; нейронная сеть; выборка; скрытый слой; скрытые нейроны; алгоритм обучения; количество эпох; погрешность

REFERENCES

1. Grishin, A. V. (2011). Neyrosetevye tekhnologii v zadachakh obnaruzheniya kompyuternykh atak. *Informatsionnye tekhnologii i vychislitelnye sistemy, 1,* 53-64. (in Ukranian).
2. Zhulkov, Y. V. (2007). *Postroenie modulnykh neyronnykh setey dlya obnaruzheniya klassov setevykh atak* (Dysertatsiia kandydata tekhnichnykh nauk). Peter the Great St. Petersburg Polytechnic University, Saint Petersburg. (in Russian)
3. Korpan, Y. V. (2015). Kompleks metodiv ta zasobiv zakhystu informatsii v kompiuternykh systemakh. *Mir nauki i innovatsiy, 3,* 31-35. (in Ukrainian)
4. Marchenko, A. A., Matvienko, S. V., & Nesteruk, F. G. (2007). Obnaruzhenie atak v sistemakh neyrosetevymi sredstvami. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 7(5),* 83-93. (in English)

doi 10.15802/stp2018/130797

5. Pakhomova, V. N. (2015). The possibilities of upgrading the computer network of information-telecommunication system of Dnieper railway. *Informatsiino-keruiuchi systemy na zaliznychnomu transporti, 5,* 32-38. (in Ukranian)
6. Piliugina, K. N. (2016). Artificial neural network approaches to intrusion detection. *Modern Scientific Researches and Innovations, 2.* Retrived from http://web.snauka.ru/issues/2016/02/63248. (in Russian)
7. Pisarenko, I. (2009). Neyrosetevye tekhnologii v bezopasnosti . *Information Security, 4.* Retrived from http://www.itsec.ru/articles2/Oborandteh/neyrosetevye-tehnologii-v-biznese. (in Russian)
8. Postarnak, D. V. (2012). Kriticheskiy analiz modeley neyronnykh setey. *Vestnik Tyumenskogo gosudarstvennogo universiteta. Fiziko-matematicheskie nauki. Informatika, 4,* 162-167. (in Russian)
9. Amini, M., Rezaeenour, J., & Hadavandi, E. (2015). Effective Intrusion Detection with a Neural Network Ensemble using Fuzzy Clustering and Stacking Combination Method. *Journal of Computing and Security, 1(4),* 293-305. (in English)
10. Amini, M. A., Rezaeenour, J., & Hadavandi, E. (2016). Neural Network Ensemble Classifier for Effective Intrusion Detection using Fuzzy Clustering and Radial Basis Function Networks. *International Journal on Artificial Intelligence Tools, 25 (02),* 1550033. doi: 10.1142/s0218213015500335. (in English)
11. Hua Yang, Tao Li, Xinlei Hu, Feng Wang, & Yang Zou. (2014). A Survey of Artificial Immune System Based Intrusion Detection. *The Scientific World Journal, 2014,* 1-11. doi: 10.1155/2014/156790. (in English)
12. Branitskiy, A., & Kotenko, I. (2015). Network attack detection based on combination of neural, immune and neuro-fuzzy classifiers. *The 18th IEEE Intern. Conf. on Computational Science and Engineering (IEEE CSE2015),* 152-159. doi: 10.1109/cse.2015.26. (in English)
13. Cannady, J. (1998). Artificial Neural Networks for Misuse Detection. *Proceedings of the 21st National Information Systems Security Conference (NISSC) (October 5–8, 1998),* 443-456. (in English)
14. *KDDCup1999Data* (1999). Retrived from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. (in English)
15. Moradi, M. (2013). System for intrusion Detection and Classification of Attacks. *Інформаційний портал університету Квінс.* Retrived from http://research.cs.queensu.ca/moradi/148-04-mm-mz.pdf. (in English)
16. Pakhomova, V. M. (2016). Network Traffic Forcasting in information-telecommunication System of Prydniprovsk Railways Based on Neuro-fuzzy Network. *Science and Transport Progress, 6(66),* 105-114. doi: 10.15802/stp2016/90485. (in English)

Prof. A. A. Kosolapov, Dr. Sc. (Tech.) (Ukraine) recommended this article to be published

Received: Jan. 09, 2018
Accessed: April 11, 2018

