ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

UDC 004.056.53:[004.7:004.032.26]

I. V. ZHUKOVYTS'KYY[1*], V. M. PAKHOMOVA[2*], D. O. OSTAPETS[3*], O. I. TSYHANOK[4*]

[1*]Dep. «Electronic Computing Machines», Dnipro National University of Railway Transport named after Academician V. Lazaryan, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail ivzhukl@ua.fm, ORCID 0000-0002-3491-5976
[2*]Dep. «Electronic Computing Machines», Dnipro National University of Railway Transport named after Academician V. Lazaryan, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail viknikpakh@gmail.com, ORCID 0000-0002-0022-099X
[3*]Dep. «Electronic Computing Machines», Dnipro National University of Railway Transport named after Academician V. Lazaryan, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail odaua@i.ua, ORCID 0000-0003-1778-7770
[4*]Dep. «Electronic Computing Machines», Dnipro National University of Railway Transport named after Academician V. Lazaryan, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail tsiganok.oleg@yandex.ua, ORCID 0000-0001-9846-7669

DETECTION OF ATTACKS ON A COMPUTER NETWORK BASED ON THE USE OF NEURAL NETWORKS COMPLEX

Purpose. The article is aimed at the development of a methodology for detecting attacks on a computer network. To achieve this goal the following tasks were solved: to develop a methodology for detecting attacks on a computer network based on an ensemble of neural networks using normalized data from the open KDD Cup 99 database; when performing machine training to identify the optimal parameters of the neural network which will provide a sufficiently high level of reliability of detection of intrusions into the computer network. **Methodology.** As an architectural solution of the attack detection module, a two-level network system is proposed, based on an ensemble of five neural networks of the multilayer perceptron type. The first neural network to determine the category of attack class (DoS, R2L, U2R, Probe) or the fact that there was no attack; other neural networks – to detect the type of attack, if any (each of these four neural networks corresponds to one class of attack and is able to identify types that belong only to this class). **Findings.** The created software model was used to study the parameters of the neural network configuration 41–1–132–5, which determines the category of the attack class on the computer network. It is determined that the optimal training speed is 0.001. The ADAM algorithm proved to be the best for optimization. The ReLU function is the most suitable activation function for the hidden layer, and the hyperbolic tangent function – for the output layer activation function. Accuracy in test and validation samples was 92.86 % and 91.03 %, respectively. **Originality.** The developed software model, which uses the Python 3.5 programming language, the integrated development environment PyCharm 2016.3 and the Tensorflow 1.2 framework, makes it possible to detect all types of attacks of DoS, U2R, R2L, Probe classes. **Practical value.** Graphical dependencies of accuracy of neural networks at various parameters are received: speed of training; activation function; optimization algorithm. The optimal parameters of neural networks have been determined, which will ensure a sufficiently high level of reliability of intrusion detection into a computer network.

Keywords: architectural solution; neural network; training speed; activation function; optimization algorithm

doi: https://doi.org/10.15802/stp2020/218318

Introduction

Efficiency of modern information systems is largely related to the problem of protecting the information processed in them. According to the Verizon 2018 Data Breach Investigations Report [14], the problem of intrusion detection is relevant. From year to year, the value of the average cost of hacking increases by 6 %. There are many algorithms for classification and detection of anomalies, each of which has its advantages and disadvantages [11]. Intrusion detection systems based on anomalies are also used to detect new types of attacks. Based on a set of queries, a model of normal behavior is formed, with which each subsequent query to the system is compared.

The capabilities of existing security systems do not allow ensuring the security of the information system at a sufficient level. The reason for this is that the process of creating attack detection systems involves a number of unsolved scientific and technical problems. Existing attack detection systems use the simplest algorithms for processing incoming information, which does not allow detecting a significant number of attacks on information systems. The main methods of detecting attacks include misuse and anomaly detection [11]. Misuse detection involves the presence of attack signatures and is based on a simple notion of coincidence of the sequence with the sample. Because the signature-based attack detection method is static, it is vulnerable to new types of attacks. To detect them, it is necessary to use systems capable of self-training in real time [6]. Creating an efficient attack detection system requires the use of qualitatively new approaches to information processing, which should be based on adaptive algorithms capable of self-training. It is known that there are two main types of implementation of intrusion detection systems based on neural networks: IDS (Intrusion-Detection System) based on a combination of expert system and neural network; IDS using neural networks (NN) as autonomous systems [6]. The most promising direction in the creation of such attack detection systems is the use of artificial intelligence. It should be noted that both large foreign commercial companies (Cisco, Computer Associates, ISS, Symantec and others) and research centers at various universities (Columbia University, Florida Institute of Technology, Purdue University, Ohio University, etc.) carry out research in this aspect.

Analysis of scientific sources. The most promising direction is IDS, built on the basis of NN: Multi Layer Perceptron (MLP); Radial Basis Function Network (RBF) and Kohonen or Self Organizing Maps (SOM). For example, in [10] only DDoS-attacks on TCP, UDP and ICMP protocols were analyzed due to their popularity among malefactors. In [13], some threats were detected in the network based on the analysis and processing of network connection parameters, which use the TCP/IP protocol stack, using a neural network configuration 19–1–25–5 (19 – is the number of input neurons; 1 – is the number of hidden layers, 25 – is the number of hidden neurons, 5 – is the number of output neurons), but other types of attacks also require research.

At the present stage, on the one hand, there are more and more works that use a combined approach to solving the problem. For example, the work [5] proposed a new ensemble classifier that uses RBF and fuzzy clustering to increase detection accuracy, reduce the number of false positives, and provide a higher detection rate for infrequent attacks. In [1] the approach with the use of neural networks, immune systems, neurofuzzy classifiers and their combinations is considered. The essence of hybrid approaches is to implement various schemes of combining basic classifiers, which allow eliminating shortcomings in their operation separately. However, an important disadvantage of such methodologies is the lack of universality of their application. In [4] to improve the efficiency of IDS, it is proposed to use the method of coincidence, based on the fact that NN with different topologies (MLP, RBF, SOM) can detect different attacks, but false positives also do not always occur on the same network packages during the analysis with the help of different types of NN. In addition, each type of neural network has its advantages and disadvantages that need to be considered or additional research conducted.

On the other hand, there are attempts to use NN at different levels. For example, in [2] a new approach to the construction of a multilevel network intrusion detection system is considered. It consists in the fact that groups of similar parameters of interconnection interaction are fed to the inputs of individual first-level modules, each of which is

doi: https://doi.org/10.15802/stp2020/218318

a hierarchical structure of several NN of different types and detects anomalies for a given group of parameters. The results of the first level modules are fed to the input of the second level solver, which makes the final decision about the presence of the attack and its classification. According to this approach, the probability of identifying known attacks was 91%, the detection of intrusions, about which there was no information during training, was 86 %. However, the developed prototype has a relatively significant probability of type II errors – 18 %, analysis and correction of the causes of these errors is promising for further study.

In addition, in large information systems (in particular, the information and telecommunication system of railway transport) there is a problem of large amounts of network traffic, this is caused by the fact that network traffic is constantly changing, and it is difficult to establish the cyclical nature of such changes. To increase the efficiency of detecting situations related to possible intrusions into the computer network, which is the basis of various information systems, recently widely used modern technologies of data mining (in particular, DataMining technologies) [8]. Network intrusion detection systems based on the anomaly detection paradigm have a high false alarm frequency, which complicates their use. To solve this weakness, in [7] it was proposed to smooth the outputs of anomaly detectors using local adaptive multifactor smoothing.

We believe that during processing a large amount of constantly changing network traffic, the use of a multi-level network system to detect various categories of MLP-based attacks using machine training (especially deep one) leads to a large number of false positives and skip attacks. Therefore, one of the approaches to solving this problem is to conduct additional research to determine the rational parameters of NN.

Purpose

Our study aims to develop a methodology for detecting attacks on a computer network. Achieving this goal involves solving the following tasks:

– develop a method of detecting attacks on a computer network based on neural networks ensemble;

– during machine training to find the optimal parameters of the neural network (MLP), which will provide a sufficiently high level of reliability of detection of intrusions into the computer network.

Methodology

Researched materials used during modeling. Attack detection system based on neural networks such as a multilayer perceptron is considered. In [3], the authors investigated two approaches to the detection of attacks on a computer network and proved the rationality of using a neural networks ensemble. In our work, the source of data for training and testing of neural networks is the KDD Cup 99 database [9], which contains more than four gigabytes of characteristics of TCP connections. The database presents the following categories of attacks: DoS, R2L, U2R, Probe. Each of these categories, in turn, is represented by several types.

DoS – network attacks aimed at creating a situation where in the attacked system denial of service takes place. Such attacks are characterized by generation of large amounts of traffic, which leads to overloading and blocking the server. There are six types of DoS-attacks: back, land, neptune, pod, smurf, teardrop.

R2L attacks are characterized by access by an unregistered user from a remote computer. There are eight types of R2L attacks: ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster.

U2R attacks involve obtaining a privilege of a local superuser (network administrator) by a registered user. There are four types of U2R attacks: buffer_overflow, loadmodule, perl, rootkit.

Probe attacks are about scanning network ports for confidential information. There are four types of Probe attacks: ipsweep, nmap, portsweep, satan.

The input vector for the attack detection system is a set of 41 TCP connection parameters, the full description of which is given in [9], examples of the description are in Table 1.

As an architectural solution of the attack detection module, five neural networks of the multilayer perceptron type are proposed: NN1 – to determine the category of the attack class (DoS, R2L, U2R, Probe) or the fact that there was no attack; NN2…NN5 – to detect the type of attack, if any

doi: https://doi.org/10.15802/stp2020/218318

(each of these four neural networks corresponds to one attack class and is able to identify types that belong only to this class). Fig. 1 shows the structure of a hypothetical complex that uses this architectural solution.

The complex contains a module for detecting network attacks, which receives connection data from network sensors and provides the result to the response module. The signal from NN1, which represents the category of the attack class, through the key «key» turns on one of the neural networks NN2…NN5, which will determine the type of attack of this class.

Table 1

Description of TCP connection parameters

No.	Title	Description	Type
1	duration	Connection duration s)	numerical
2	protocol_type	type of protocol (TCP, UDP, etc.)	symbolic
3	service	end network service (HTTP, Telnet, etc.)	symbolic
4	flag	connection status (normal or false)	symbolic
5	src_byte	the number of bytes transmitted from transmitter to receiver	numerical
6	dst_byte	the number of bytes transmitted from the receiver to the transmitter	numerical
7	land	equality of hosts/ports of receiver and transmitter	symbolic
8	wrong_fragment	the number of «wrong» fragments	numerical
9	hot	number of hot indicators	numerical
10	num_failed_logins	the number of failed login attempts	numerical
…	…	…	…
41	dst_host_srv_rerror_rate	the percentage of REJ error connections for the customer service	numerical

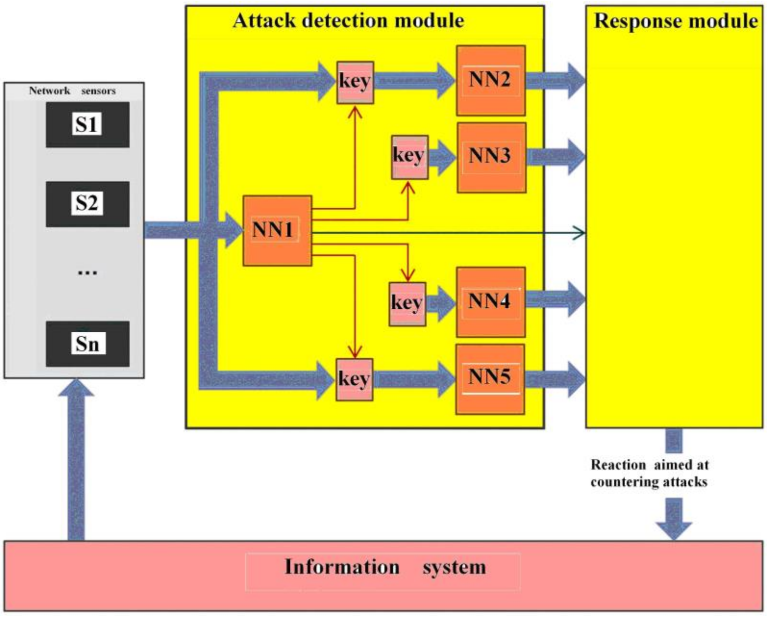


Fig. 1. Structure of hypothetical complex

doi: https://doi.org/10.15802/stp2020/218318

Neural network to identify the category of attack class or determine the fact of its absence. The input vector is 41 TCP connection parameters, the output is a vector of five values, four of which represent the attack class, and the fifth is the regular connection. This is the so-called one-hot vector – a vector whose components are equal to zero, except for one, which is equal to one. This component will indicate the neural network-defined attack class or the normal connection, if there was no attack. The number of neurons in the hidden layer of the multilayer perceptron can be determined by a known formula, which is a consequence of the Kolmogorov–Arnold–Hecht–Nielsen theorem:

$$\frac{N_y Q}{1+log_2(Q)} \le N_w \le$$

$$\le N_y(\frac{Q}{N_x}+1)(N_x+N_y+1)+N_y, \quad (1)$$

where N_y – the length of the output signal; Q – the number of elements of the set of training examples; N_w – the required number of synaptic connections; N_x – dimension of the input signal.

Having estimated the required number of synaptic connections, you can calculate the required number of neurons in the hidden layer:

$$N=\frac{N_w}{N_x+N_y}. \quad (2)$$

The values N_x, N_y, Q are 41, 5 and 1024 records, respectively. It is found out that $466 \le N_w \le 6110$; $11 \le N \le 132$. Let us take a larger value – 132 neurons, then NN1 will have the configuration 41–1–132–5, which is presented in Fig. 2, where 41 is the number of input neurons, 1 is the number of hidden layers, 132 is the number of hidden neurons, and 5 is the number of output neurons.

By the way, the architectures of all other NN are similar and differ only in the number of neurons in different layers, the results of the calculations are summarized in Table 2.

General characteristics of the software model. Python 3.5 programming language, PyCharm 2016.3 integrated development environment and Tensorflow 1.2 framework were used to create the software model [12].

The software allows you to build a multilayer perceptron (add layers of a given length), train and test the neural model. In addition, you can set such parameters as the convergence coefficient, the number of training epochs, size of the data portion for the training step. The ADAM optimizer is taken as the optimization function. In the program, you can calculate the difference between the responses of the neural network and reference networks and obtain the values of accuracy.

The Tensorboard visualization tool was used to analyze neural models.

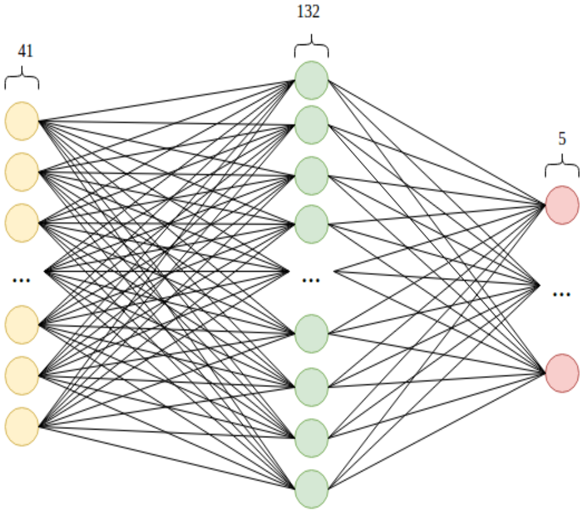


Fig. 2. NN configurations 41–1–132–5
to identify the attack class category

Findings

As an example, the analysis of NN1, which defines the category of attack class. For NN1 of configuration 41–1–132–5, the length of the training, test and validation sample is 1024, 294 and 156 examples, respectively.

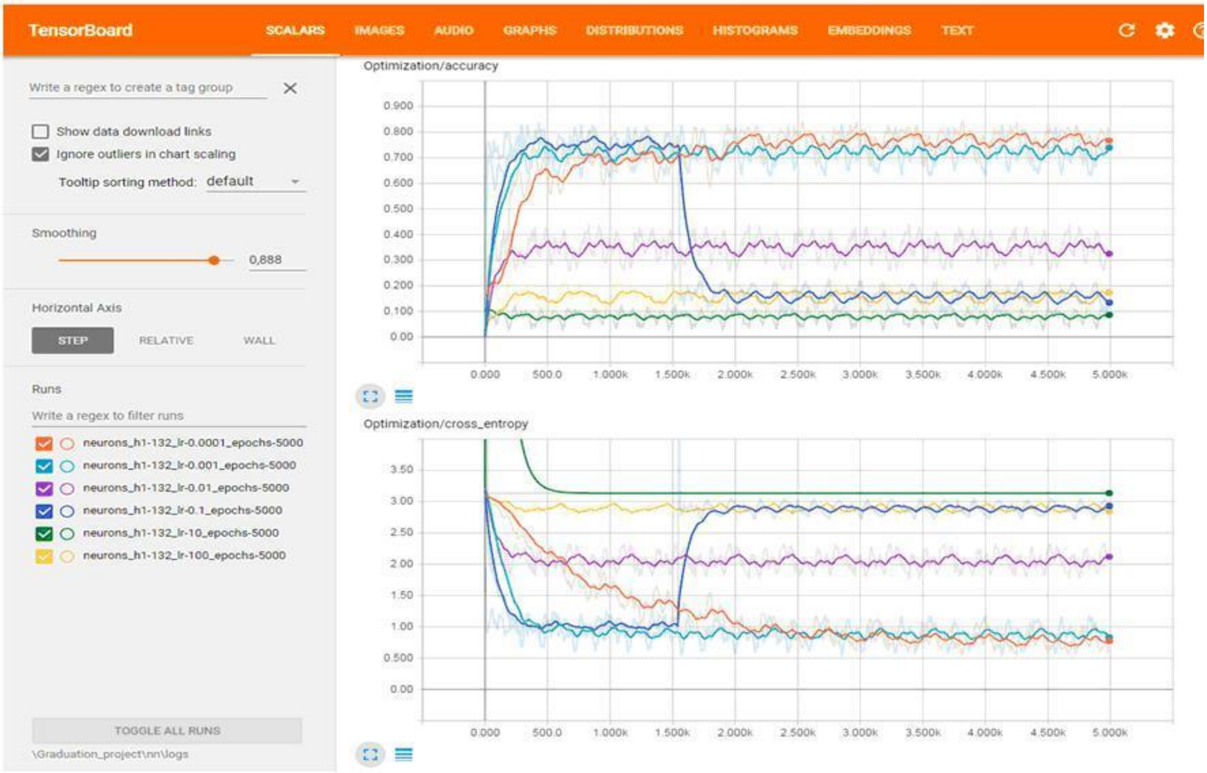
The obtained graphs of accuracy and error of training of NN1 of configuration 41–1–132–5 by iterations for different training speeds are presented in Fig. 3. The total number of training iterations is 5000; training data come in 100 lines per iteration. The Smoothing parameter is responsible for approximating the curves on the graphs, which allows you to estimate the rate of their growth or decline. Bright colors indicate approximated graphics, pale – their true view.

doi: https://doi.org/10.15802/stp2020/218318

Table 2

NN parameters and configuration

NN	Appointment	NN parameters					Configuration
		N_x	N_y	Q	N_w	N	
NN1	Detection of attack class category	41	5	1 024	$466 \le N_w \le 6110$	$11 \le N \le 132$	41–1–132–5
NN2	Detection of the attack type of DoS category	41	7	1 024	$559 \le N_w \le 7\,487$	$12 \le N \le 160$	41–1–160–5
NN3	Detection of the attack type of R2L category	41	9	512	$410 \le N_w \le 5\,404$	$9 \le N \le 111$	41–1–111–5
NN4	Detection of the attack type of U2R category	41	5	32	$22 \le N_w \le 332$	$1 \le N \le 8$	41–1–8–5
NN5	Detection of the attack type of Probe category	41	5	1 024	$373 \le N_w \le 4\,787$	$9 \le N \le 107$	41–1–107–5



Write a regex to create a tag group

☐ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0,888

Horizontal Axis: [✓ STEP] RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ neurons_h1-132_lr-0.0001_epochs-5000
☑ ○ neurons_h1-132_lr-0.001_epochs-5000
☑ ○ neurons_h1-132_lr-0.01_epochs-5000
☑ ○ neurons_h1-132_lr-0.1_epochs-5000
☑ ○ neurons_h1-132_lr-10_epochs-5000
☑ ○ neurons_h1-132_lr-100_epochs-5000

\Graduation_project\nn\logs

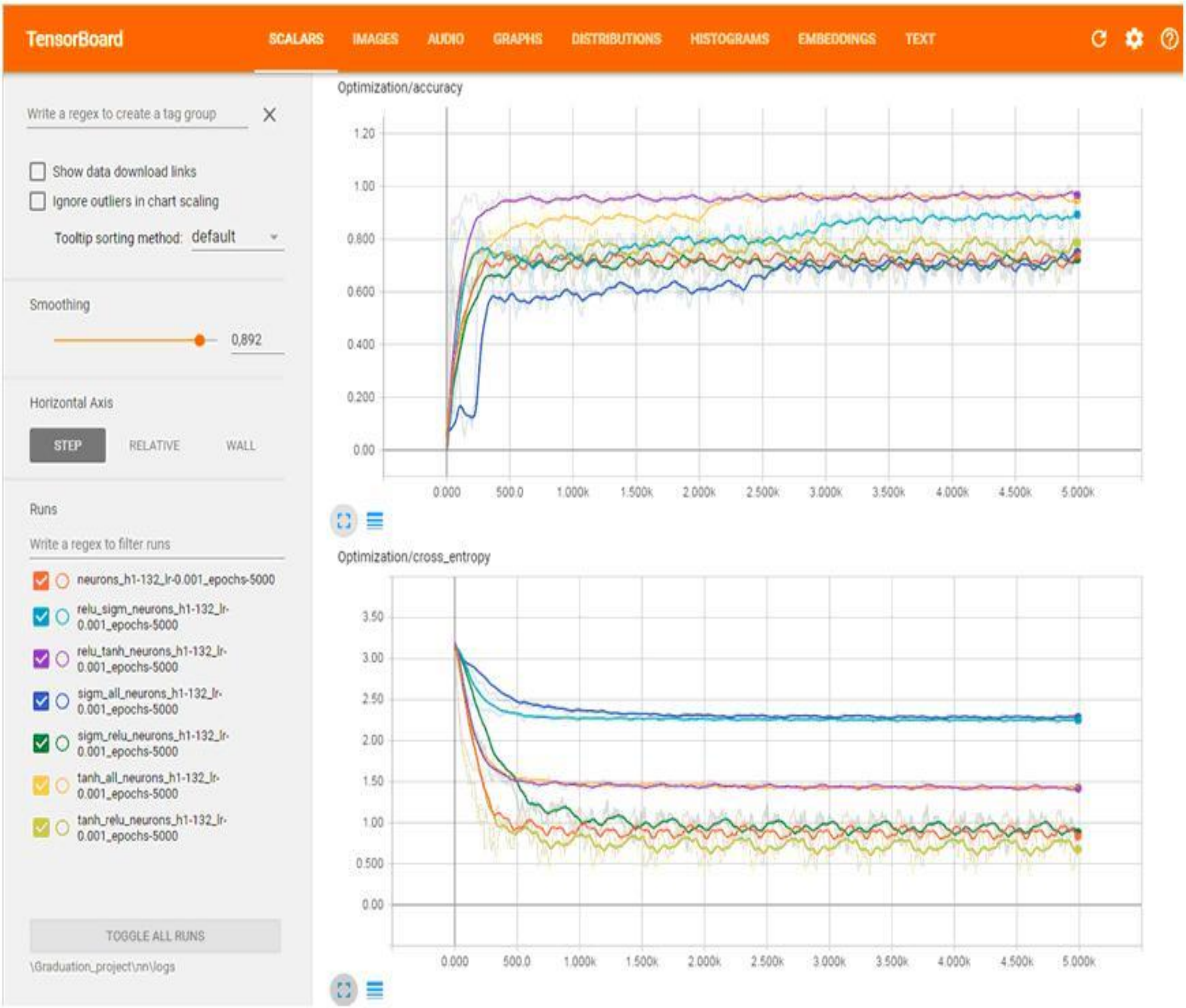
neurons_h1-N_lr-M_epochs-P:
N – the number of neurons in the hidden layer, *M* – training speed, *P* – number of training iterations

Fig. 3. Accuracy and error of NN training 41–1–132–5 by iterations (for different training speeds)

doi: https://doi.org/10.15802/stp2020/218318

As can be seen from Fig. 3, at a training speed of 0.1, the accuracy dropped sharply between the second and third thousand iterations, which may be the result of a poor sample. However, at a training speed of 0.0001 the result is satisfactory, and at a training speed of 0.001 – the result is the best.

The obtained graphs of accuracy and training error of NN1 of configuration 41–1–132–5 by iterations for different activation functions are presented in Fig. 4. The training speed is 0.001, the number of training iterations is 5000, the length of the data portion on each iteration is 100 lines.



Write a regex to create a tag group

□ Show data download links

□ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0,892

Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ neurons_h1-132_lr-0.001_epochs-5000
☑ ○ relu_sigm_neurons_h1-132_lr-0.001_epochs-5000
☑ ○ relu_tanh_neurons_h1-132_lr-0.001_epochs-5000
☑ ○ sigm_all_neurons_h1-132_lr-0.001_epochs-5000
☑ ○ sigm_relu_neurons_h1-132_lr-0.001_epochs-5000
☑ ○ tanh_all_neurons_h1-132_lr-0.001_epochs-5000
☑ ○ tanh_relu_neurons_h1-132_lr-0.001_epochs-5000

\Graduation_project\nn\logs

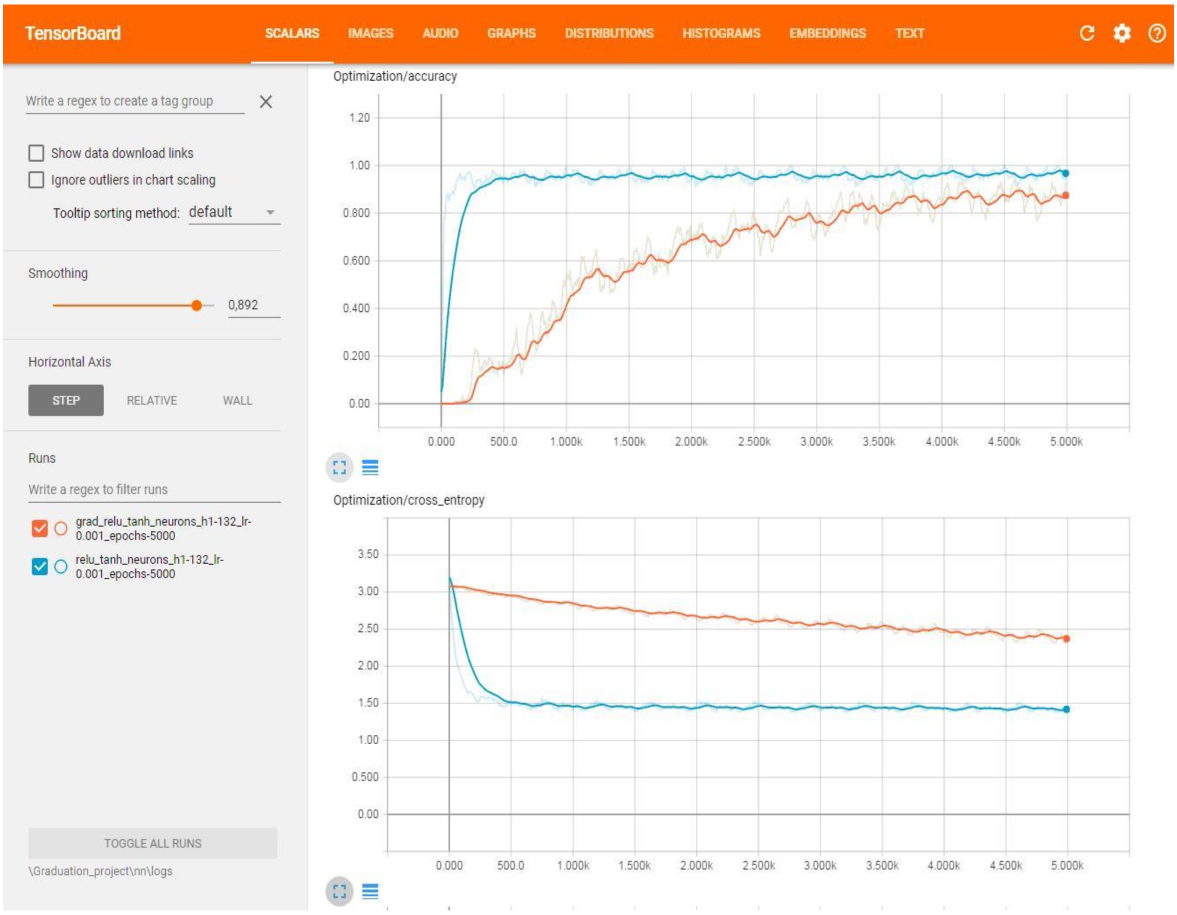
[f1_][f2_]neurons_h1-132_lr-0,001_epochs-5000:
*f*1, *f*2 – hidden and output layer activation functions; by default – relu; all – *f*2 repeats*f*1

Fig. 4. Accuracy and error of NN 41–1–132–5 training by iterations (for different activation functions)

Fig. 4 shows that the best results (less time spent on NN training) are achieved when using the ReLU function in the hidden layer and the hyperbolic tangent in the output layer.

The obtained graphs of accuracy and error of training NN 41–1–132–5 by iterations for different optimization algorithms are presented in Fig. 5. Parameters of NN1: training speed – 0.001, number of iterations – 5000, data portion length – 100, hidden layer activation function – ReLU, output layer activation function – hyperbolic tangent.

doi: https://doi.org/10.15802/stp2020/218318



Write a regex to create a tag group ✕

□ Show data download links

□ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

0,892

Horizontal Axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ grad_relu_tanh_neurons_h1-132_lr-0.001_epochs-5000

☑ ○ relu_tanh_neurons_h1-132_lr-0.001_epochs-5000

\Graduation_project\nn\logs

grad_relu_tanh_neurons_h1-132_lr-0,001_epochs-5000 – the optimization algorithm is a gradient descent,
relu_tanh_neurons_h1-132_lr-0,001_epochs-5000 – the optimization algorithm is ADAM

Fig. 5. Accuracy and error of NN 41–1–132–5 training by iterations (for different optimization algorithms)

Fig. 5 shows that for NN1 the ADAM algorithm works faster.

It was determined that NN1 of configuration 41–1–132–5 provides the best accuracy value at a training speed of 0.001 and requires the least training time when using the semilinear ReLU activation function in the hidden layer and the hyperbolic tangent in the output layer; ADAM algorithm compared to gradient descent algorithm works faster, gives higher accuracy and lower error.

According to the results of the study of NN1 of configuration 41–1–132–5 to identify the category of attack class the following parameters were determined: learning speed – 0.001; number of iterations – 5000; data portion length – 100; activation function in the hidden layer – ReLU; activation function in the output layer – hyperbolic tangent; optimization algorithm – ADAM, for which the accuracy of the test and validation samples was 92.86 and 91.03 %, respectively.

The simulation results on other neural networks (attack detection accuracy) are summarized in Table 3.

Table 3

NN simulation results

NN	NN1	NN2	NN3	NN4	NN5
Configuration	41–1–132–5	41–1–160–5	41–1–111–5	41–1–8–5	41–1–107–5
Accuracy, %	91.03	98.93	94.77	–	97.35

doi: https://doi.org/10.15802/stp2020/218318

The table shows that the best result is achieved when determining the type of attacks of the DoS and Probe classes, slightly worse – for the R2L class. For the U2R class, it was not possible to configure the NN4 neural network to obtain acceptable results. This is due to the small number of records (52 in total) in the KDD Cup 99 database that belong to the U2R class.

Originality and practical value

In our work, the detection of network attacks was carried out using the apparatus of neural networks (multilayer perceptron), as in other works [10, 13], which is not a contradiction to those works [1, 4, 5], where a hybrid (immune mechanisms and SOM; neural, immune and neuro-fuzzy classifiers) or combined approach used based on different types of neural networks (MLP and neural fuzzy network; multiple SOM and neural fuzzy network; MLP, RBF and SOM). Herewith, in our work all types of attacks of DoS classes are investigated; U2R; R2L; Probe, not individual, as in [10, 13].

We believe that the use of a multi-layer perceptron as a mathematical apparatus is appropriate and sufficient. For example, although the RBF network is trained faster than the MLP network, it is necessary to determine the number of radial elements, location of their centers and deviation values, the RBF model requires slightly more elements, i.e. will run slower and requires more memory than the MLP model.

Processing a large amount of constantly changing network traffic, based on MLP using machine training (especially deep) leads to a large number of false positives and skip attacks, which requires additional research using DataMining technology [8]. Thus, in particular, in our work on the software model it is determined that the ADAM optimization algorithm works faster than the gradient descent algorithm. It gives higher accuracy and lower error; this cannot be a contradiction to the use of other means (in particular, local adaptive multifactor smoothing, proposed in [7]).

The study used a multilevel (namely two-level) approach to building a network system for detecting intrusions into a computer network: determining the category of attack class (first level); assigning the type of attack to the appropriate class (second level), which is also not a contradiction to the modular approach in [2]. But the probability of error of the second kind (the number of skip of attacks) in our work is about 10 vs. 18 % for the modular approach, which is implemented in [2], which is 1.8 times better.

Conclusions

1. When processing a large amount of constantly changing network traffic, it is appropriate to use a two-level network system based on five neural networks of the following configurations: 41–1–132–5 to determine the category of attack class at the first level, as well as 41–1–160–7, 41–1–8–5, 41–1–111–9, 41–1–107–5 to detect the type of attack from the DoS classes (back, land, neptune, pod, smurf, teardrop), U2R (buffer_overflow, loadmodule, perl, rootkit), R2L (ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warez-master), Probe (ipsweep, nmap, portsweep, sa-tan) respectively at the second level. The training data are taken from the open KDD Cup 99 database, which stores a large number of characteristics of TCP connections. The Google TensorFlow machine training framework was chosen to build all neural networks because of its flexibility and speed.

2. We conducted a study of the parameters of the neural network configuration 41–1–132–5, which determines the category of the attack class on the computer network. It is determined that the optimal training speed is 0.001. The ADAM algorithm proved to be the best for optimization. As a function of activation for the hidden layer, the ReLU function is the most suitable, for the activation function of the output layer – the hyperbolic tangent function. The accuracy in the test and validation samples was 92.86 and 91.03 %, respectively. The probability of a type II error is about 10%.

3. The study showed that for training a neural network of 41–1–8–5 configuration, which determines the type of attack of the U2R class, the available training sample is not enough.

doi: https://doi.org/10.15802/stp2020/218318

LIST OF REFERENCE LINKS

1. Браницкий А. А. *Обнаружение аномальных сетевых соединений на основе гибридизации методов вычислительного интеллекта* : автореф. дис. ... канд. техн. наук. Санкт-Петербург, 2018. 18 с.
2. Жульков Е. В. *Построение модульных нейронных сетей для обнаружения классов сетевых атак* : автореф. дис. ... канд. техн. наук. Санкт-Петербург, 2007. 16 с.
3. Пахомова В. М., Коннов М. С. Дослідження двох підходів до виявлення мережних атак із використанням нейромережної технології. *Наука та прогрес транспорту*. 2020. № 3 (87). С. 81–93. DOI: https://doi.org/10.15802/stp2020/208233
4. Фролов П. В., Чухраев И. В., Гришанов К. М. Применение искусственных нейронных сетей в системах обнаружения вторжений. *Системный администратор*. 2018. № 9 (190). URL: http://samag.ru/archive/article/3724 (дата звернення: 04.09.2020).
5. Amini M., Rezaeenour J., Hadavandi E. A Neural Network Ensemble Classifier for Effective Intrusion Detection using Fuzzy Clustering and Radial Basis Function Networks. *International Journal on Artificial Intelligence Tools*. 2016. Vol. 25. Iss. 02. P. 1–32. DOI: https://doi.org/10.1142/s0218213015500335
6. Esteban J. A New GHSOM Model applied to network security. *Artificial Neural Networks-ICANN 2008*. 2008. P. 680–689.
7. Grill M., Pevný T., Rehak M. Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences*. 2017. Vol. 83. Iss. 1. P. 43–57. DOI: https://doi.org/10.1016/j.jcss.2016.03.007
8. Hadi, A. A. A. Performance Analysis of Big Data Intrusion Detection System over Random Forest Algorithm. *International Journal of Applied Engineering Research*. 2018. Vol. 13, No. 2. P. 1520–1527.
9. KDD Cup 1999 Data. URL: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (дата звернення: 04.09.2020).
10. Saied A., Overill R. E., Radzik T. Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing*. 2016. Vol. 172. P. 385–393. DOI: https://doi.org/10.1016/j.neucom.2015.04.101
11. Sikos L. F. *AI in Cybersecurity*. New York : Springer, 2018. 205 p.
12. TensorFlow. URL: http://www.tensorflow.org (дата звернення: 04.09.2020).
13. Zhukovyts'kyy I. V., Pakhomova V. M. Identifying threats in computer network based on multilayer neural network. *Наука та прогрес транспорту*. 2018. № 2 (74). P. 114–123. DOI: https://doi.org/10.15802/stp2018/130797
14. 2018 Data Breach Investigations Report. URL: https://enterprise.verizon.com/resources/reports/DBIR_2018_Report.pdf (дата звернення: 04.09.2020).

І. В. ЖУКОВИЦЬКИЙ[1*], В. М. ПАХОМОВА[2*], Д. О. ОСТАПЕЦЬ[3*], О. І. ЦИГАНОК[4*]

[1*]Каф. «Електронні обчислювальні машини», Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (056) 373 15 89, ел. пошта ivzhukl@ua.fm, ORCID 0000-0002-3491-5976
[2*]Каф. «Електронні обчислювальні машини», Дніпровський 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
[3*]Каф. «Електронні обчислювальні машини», Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (056) 373 15 89, ел. пошта odaua@i.ua, ORCID 0000-0003-1778-7770
[4*]Каф. «Електронні обчислювальні машини», Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (056) 373 15 89, ел. пошта tsiganok.oleg@yandex.ua, ORCID 0000-0001-9846-7669

ВИЯВЛЕННЯ АТАК НА КОМП'ЮТЕРНУ МЕРЕЖУ НА ОСНОВІ ВИКОРИСТАННЯ КОМПЛЕКСУ НЕЙРОННИХ МЕРЕЖ

Мета. За основну мету дослідження ми ставимо розвиток методики визначення атак на комп'ютерну мережу. Досягнення поставленої мети передбачає вирішення таких завдань: розробити методику виявлення атак на комп'ютерну мережу на основі ансамблю нейронних мереж із використанням нормалізованих да-

doi: https://doi.org/10.15802/stp2020/218318

них відкритої бази KDDCup99; під час виконання машинного навчання виявити оптимальні параметри нейронної мережі, що забезпечить достатньо високий рівень достовірності виявлення вторгнень у комп'ютерну мережу. **Методика.** Як архітектурне рішення модуля виявлення атак запропоновано дворівневу мережну систему, основу якої складає ансамбль із п'яти нейронних мереж типу багатошарового персептрона: перша нейронна мережа – для визначення категорії класу атаки (DoS, R2L, U2R, Probe) або факту того, що атаки не було; інші нейронні мережі – для виявлення типу атаки, якщо така мала місце (кожна з цих чотирьох нейронних мереж відповідає одному класу атаки і вміє визначати типи, що належать тільки цьому класу). **Результати.** На створеній програмній моделі проведено дослідження параметрів нейронної мережі конфігурації 41–1–132–5, яка визначає категорію класу атаки на комп'ютерну мережу. Встановлено, що оптимальна швидкість навчання дорівнює 0,001. Для оптимізації найкраще себе показав алгоритм ADAM. Як функція активації для прихованого шару найбільше підходить функція ReLU, для функції активації вихідного шару – функція гіперболічного тангенса. Точність на тестовій та валідаційній вибірках склала 92,86 та 91,03 % відповідно. **Наукова новизна**. Розроблена програмна модель, для якої використана мова програмування Python 3.5, інтегроване середовище розробки PyCharm 2016.3 та фреймворк Tensorflow 1.2, дає можливість виявляти всі типи атак класів DoS,U2R, R2L, Probe. **Практична значимість**. Отримано графічні залежності точності нейронних мереж за різних параметрів: швидкості навчання; активаційної функції; алгоритму оптимізації. Визначено оптимальні параметри нейронних мереж, що забезпечать достатньо високий рівень достовірності виявлення вторгнень у комп'ютерну мережу.

Ключові слова: архітектурне рішення; нейронна мережа; швидкість навчання; функція активації; алгоритм оптимізації

И. В. ЖУКОВИЦКИЙ[1*], В. Н. ПАХОМОВА[2*], Д. А. ОСТАПЕЦ[3*], О. И. ЦЫГАНОК[4*]

[1*]Каф. «Электронные вычислительные машины», Днипровский национальный университет железнодорожного транспорта имени академика В. Лазаряна, ул. Лазаряна, 2, Днипро, Украина, 49010, тел. +38 (056) 373 15 89,
эл. почта ivzhukl@ua.fm, ORCID 0000-0002-3491-5976
[2*]Каф. «Электронные вычислительные машины», Днипровский национальный университет железнодорожного транспорта имени академика В. Лазаряна, ул. Лазаряна, 2, Днипро, Украина, 49010, тел. +38 (056) 373 15 89,
эл. почта viknikpakh@gmail.com, ORCID 0000-0002-0022-099X
[3*]Каф. «Электронные вычислительные машины», Днипровский национальный университет железнодорожного транспорта имени академика В. Лазаряна, ул. Лазаряна, 2, Днипро, Украина, 49010, тел. +38 (056) 373 15 89,
эл. почта odaua@i.ua, ORCID 0000-0003-1778-7770
[4*]Каф. «Электронные вычислительные машины», Днипровский национальный университет железнодорожного транспорта имени академика В. Лазаряна, ул. Лазаряна, 2, Днипро, Украина, 49010, тел. +38 (056) 373 15 89,
эл. почта tsiganok.oleg@yandex.ua, ORCID 0000-0001-9846-7669

ОБНАРУЖЕНИЕ АТАК НА КОМПЬЮТЕРНУЮ СЕТЬ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ КОМПЛЕКСА НЕЙРОННЫХ СЕТЕЙ

Цель. В качестве основной цели исследования поставлено развитие методики определения атак на компьютерную сеть. Достижение поставленной цели предусматривает решение следующих задач: разработать методику выявления атак на компьютерную сеть на основе ансамбля нейронных сетей с использованием нормализованных данных открытой базы KDDCup99; при выполнении машинного обучения выявить оптимальные параметры нейронной сети, что обеспечит достаточно высокий уровень достоверности обнаружения вторжений в компьютерную сеть. **Методика.** В качестве архитектурного решения модуля обнаружения атак предложено двухуровневую сетевую систему, основу которой составляет ансамбль из пяти нейронных сетей типа многослойного персептрона: первая нейронная сеть – для определения категории класса атаки (DoS, R2L, U2R, Probe) или факта того, что атаки не было; другие нейронные сети – для выявления типа атаки, если таковая имела место (каждая из этих четырех нейронных сетей соответствует одному классу атаки и умеет определять типы, принадлежащих только этому классу). **Результаты.** На созданной программной модели проведено исследование параметров нейронной сети конфигурации 41–1–132–5, которая определяет категорию класса атаки на компьютерную сеть. Установлено, что оптимальная скорость обучения равна 0,001. Для оптимизации лучше всего себя показал алгоритм ADAM. В качестве функции активации для скрытого слоя более всего подходит функция ReLU, для функции активации выходного слоя – функция гиперболического тангенса. Точность на тестовой и валидационной выборках составила 92,86

doi: https://doi.org/10.15802/stp2020/218318

и 91,03 % соответственно. **Научная новизна.** Разработанная программная модель, для которой использован язык программирования Python 3.5, интегрированная среда разработки PyCharm 2016.3 и фреймворк Tensorflow 1.2, дает возможность обнаруживать все типы атак классов DoS, U2R, R2L, Probe. **Практическая значимость.** Получены графические зависимости точности нейронных сетей при различных параметрах: скорости обучения; активационной функции; алгоритма оптимизации. Определены оптимальные параметры нейронных сетей, которые обеспечат достаточно высокий уровень достоверности обнаружения вторжений в компьютерную сеть.

Ключевые слова: архитектурное решение; нейронная сеть; скорость обучения; функция активации; алгоритм оптимизации

REFERENCES

1. Branitskiy, A. A. (2018). *Obnaruzhenie anomalnykh setevykh soedineniy na osnove gibridizatsii metodov vychislitelnogo intellekta* (Extended abstract of PhD dissertation). St. Petersburg, Russia. (in Russian)
2. Zhulkov, Ye. V. (2007). *Postroenie modulnykh neyronnykh setey dlya obnaruzheniya klassov setevykh atak* (Extended abstract of PhD dissertation). St. Petersburg, Russia. (in Russian)
3. Pakhomova, V. M., & Konnov, M. S. (2020). Research of two approaches to detect network attacks using neural network technologies. *Science and Transport Progress, 3*(87), 81-93. DOI: https://doi.org/10.15802/stp2020/208233 (in Ukrainian)
4. Frolov, P. V., Chukhraev, I. V., & Grishanov, K. M. (2018). Application of artificial neural networks in intrusion detection systems. *System administrator, 9*(190). Retrieved from samag.ru/archve/article/3724 (in Russian)
5. Amini, M., Rezaeenour, J., & Hadavandi, E. (2016). A Neural Network Ensemble Classifier for Effective Intrusion Detection Using Fuzzy Clustering and Radial Basis Function Networks. *International Journal on Artificial Intelligence Tools, 25*(02), 1-32. DOI: https://doi.org/10.1142/s0218213015500335 (in English)
6. Esteban, J. (2008). A New GHSOM Model applied to network security. *Artificial Neural Networks-ICANN 2008* (pp. 680-689). (in English)
7. Grill, M., Pevný, T., & Rehak, M. (2017). Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences, 83*(1), 43-57. DOI: https://doi.org/10.1016/j.jcss.2016.03.007 (in English)
8. Hadi, A. A. A. (2018). Performance Analysis of Big Data Intrusion Detection System over Random Forest Algorithm. *International Journal of Applied Engineering Research*, *13*(2), 1520-1527 (in English)
9. KDD Cup 1999 Data. Retrieved from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (in English)
10. Saied, A., Overill, R. E., & Radzik, T. (2016). Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing, 172*, 385-393. DOI: https://doi.org/10.1016/j.neucom.2015.04.101 (in English)
11. Sikos, L. F. (2018). *AI in Cybersecurity*. New York: Springer. (in English)
12. TensorFlow. Retrieved from http://www.tensorflow.org (in English)
13. Zhukovyts'kyy, I. V., & Pakhomova, V. M. (2018). Identifying threats in computer network based on multilayer neural network. *Science and Transport Progress, 2*(74), 114-123. DOI: https://doi.org/10.15802/stp2018/130797 (in English)
14. 2018 Data Breach Investigations Report. Retrieved from https://enterprise.verizon.com/resources/reports/DBIR_2018_Report.pdf (in English)

Received: May 28, 2020
Accepted: September 28, 2020

doi: https://doi.org/10.15802/stp2020/218318