https://www.sworldjournal.com/index.php/swj/article/view/swj26-00-012
DOI: 10.30888/2663-5712.2024-26-00-012

UDC 004.056.53:[004.7:004.032.26]

DETECTION OF U2R ATTACKS BY MEANS OF A MULTILAYER NEURAL NETWORK

ВИЯВЛЕННЯ U2R АТАК ЗАСОБАМИ БАГАТОШАРОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Victoria Pakhomova / Вікторія Пахомова
c.t.s., as.prof. / к.т.н., доц.
ORCID: 0000-0002-0022-099X
Vladyslav Mostynets / Владислав Мостинець
student / студент
ORCID: 0009-0009-4022-7983
Ukrainian State University of Science and Technology, Dnipro, Lazaryan, 2, 49010
Український державний університет науки і технологій, Дніпро, Лазаряна, 2, 49010

Abstract. *As a research method, multi layer neural network (MLNN) configurations 41-1-X-4 were used, where 41 is the number of input neurons; 1 – the number of hidden layers; X – the number of hidden neurons; 4 – the number of resultant neurons created using the Neural Network Toolbox of the MatLAB system, to detect U2R network attacks: y1 – Rootkit attack, y2 –Buffer overflow attack, y3 – Loadmodule attack, y4 – No attack. Using the open database of NSL-KDD network traffic parameters on the created MLNN, a study of its error and number of* epochs *at different number of hidden neurons (25, 35 and 45 was carried out using different training algorithms: Levenberg-Marquardt; Bayesian Regularization; Scaled Conjugate Gradient. It is determined that the smallest value of the MLNN error was based on the use of the hyperbolic tangent as a function of activating a hidden layer according by the Levenberg-Marquardt training algorithm, and it is enough to have 25 hidden neurons. An assessment of the quality of detection of U2R attacks on MLNN configuration 41-1-25-4 at its optimal parameters was carried out. It is determined that errors of the first and second kind are 9 % and 10 %, respectively.*

Keywords: *U2R, traffic, NSL-KDD, MLNN, hyperbolic tangent function, MLNN error, error of the first kind, error of the second kind.*

Introduction

Formulation of the problem. The modern development of information technology is increasing the number and variety of network attacks every year, which poses a threat to computer networks. Detection of such attacks using neuronetwork technology is extremely important at the present stage.

Analysis of the latest research. One of the most common attacks is U2R (User to Root) network attacks, which allow attackers to gain administrator rights. To detect U2R network attacks, radial basis function network (RBF) is proposed in [2], and Kohonen network self-organizing map (SOM) in [3], but there is also a multi layer neural network (MLNN), which requires additional research.

The purpose of the article is development of a methodology for determining the U2R attacks by means of neural networks. In accordance with the purpose, the following tasks are set: creation of a MLNN; study of optimal parameters on the created MLNN; determination of error of the first and error of second kind on the created MLNN.

1. Statement of the problem and mathematical apparatus

U2R network attacks are system attacks in which a hacker starts a system with a normal user account and tries to abuse vulnerabilities in the system to gain superuser privileges. This type of attack is divided into the following classes: Buffer_overflow, Loadmodule, Perl (but it was not considered due to a lack of examples), Rootkit. As a mathematical apparatus, the MLNN configuration 41-1-X-4, where 4 is the number of input neurons; 1 – the total number of hidden layers; X – the total number of neurons of hidden layer; 4 – the number of resultant neurons (y1 – Rootkit attack, y2 – Buffer_overflow attack, y3 – Loadmodule attack, y4 – No attack), and the structure of which is shown in Figure 1.

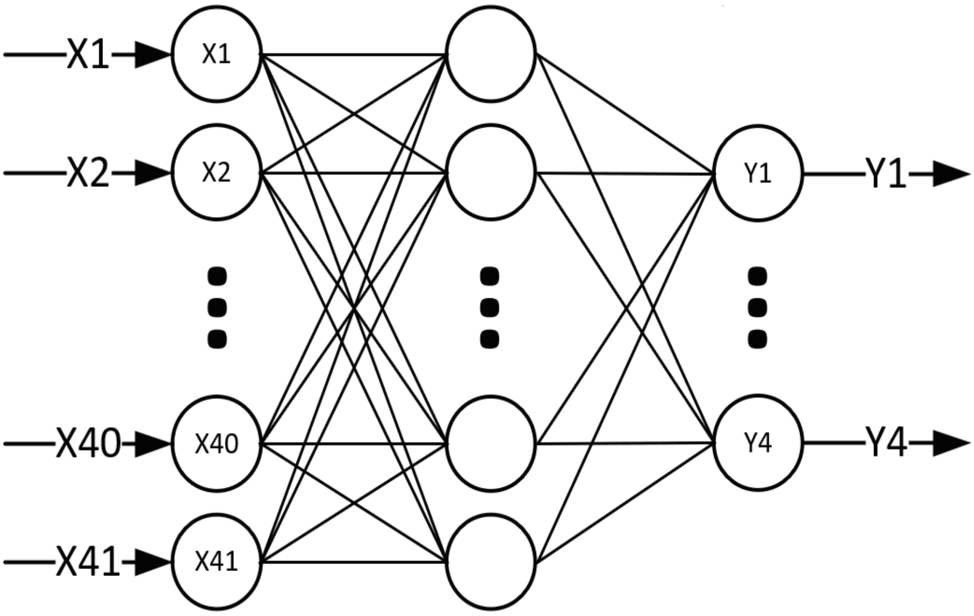

Figure 1 – MLNN configuration 41-1-X-4

The first layer of MLNN has X1…X41 neurons (these are the parameters of network traffic), which are summarized in Table 1.

Table 1 – List of neurons of the first layer of MLNN

Нейрон	*Назва*	*Опис*
1	***2***	***3***
X1	duration	connection duration
X2	protocol_type	protocol type
X3	service	service used in the connection
X4	flag	checkbox indicating the status of the package
X5	src_bytes	number of bytes sent from source to destination
X6	dst_bytes	number of bytes sent from destination to source
X7	land	indicates whether the connection is a Land attack (1 - yes, 0 - no)
X8	wrong_fragment	number of irregular snippets
X9	urgent	number of high priority packets
X10	hot	number of "hot" (frequently visited) destinations
X11	num_failed_logins	number of unsuccessful login attempts
X12	logged_in	checkbox indicating whether or not you have logged in system

X13	num_compromised	number of compromised systems associated with the package
X14	root_shell	indicates whether it has been installed root shell
X15	su_attempted	checkbox indicating whether or not the privilege elevation command has been attempted
X16	num_root	number of teams from root
X17	num_file_creations	number of files created
X18	num_shells	number of skins performed during a session
X19	num_access_files	number of files with access
X20	num_outbound_cmds	number of outgoing commands
X21	is_host_login	checkbox indicating whether or not you have logged in as a host
X22	is_guest_login	checkbox indicating whether or not you signed in as a guest
X23	count	number of last-second connections to the host
X24	srv_count	number of connections to one service in the last second
X25	serror_rate	frequency of connections with errors (service errors)
X26	srv_serror_rate	frequency of connections to the same service with errors (service errors)
X27	rerror_rate	frequency of connections with errors (system errors)
X28	srv_rerror_rate	frequency of connections to the same service with errors (system errors)
X29	same_srv_rate	frequency of connections to one service with the same type of service
X30	diff_srv_rate	frequency of connections to different services
X31	srv_diff_host_rate	frequency of connections to different hosts for the same service
X32	dst_host_count	number of unique hosts to which the connection took place
X33	dst_host_srv_count	number of unique hosts to which connections to one service have occurred
X34	dst_host_same_srv_rate	frequency of connections to one service on one host
X35	dst_host_diff_srv_rate	frequency of connections to different services on the same host
X36	dst_host_same_src_port_rate	frequency of connections from one source port to one destination port
X37	dst_host_srv_diff_host_rate	frequency of connections to different hosts for the same service on the same host
X38	dst_host_serror_rate	frequency of connections to a single host with errors (service errors)
X39	dst_host_srv_serror_rate	frequency of connections to one service on one host with errors (service errors)

X40	dst_host_rerror_rate	frequency of connections to a single host with errors (system errors)
X41	dst_host_srv_rerror_rate	frequency of connections to one service on one host with errors (system errors)

2. Sample preparation

Based on an open database NSL-KDD [1] a sample of 40 examples (10 examples for each case) was compiled, a fragment of which is shown in Figure 2. The sample fragment for neurons of the result layer of MLNN is presented in Figure 3.

98	0	0	0	621	8356	0	0	1	1	0	1	5	1	0	14	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	4	0.02	0.02	0	0	0	0	0	0
708	0	0	0	1727	24080	0	0	0	0	0	1	6	0	0	7	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	3	0.01	0.02	0	0	0	0	0	0
0	1	2	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	2	2	1	0	0.50	0	0	0	0	0
61	0	0	0	294	3929	0	0	0	0	0	1	0	1	0	4	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	4	0.02	0.02	0	0	0	0.25	0.73	0.25
0	1	2	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	1	0	0.02	0	0	0	0	0	0
0	0	1	0	0	5696	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	81	1	0	1	0.02	0	0	0	0
0	0	1	0	0	5828	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	2	2	1	0	1	0	0	0	0	0
179	0	0	0	1559	2855	0	0	0	2	0	1	4	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	2	2	1	0	0.50	0	0	0	0	0
113	0	0	0	6274	16771	0	0	0	5	0	1	2	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0
169	0	0	0	1567	2857	0	0	0	3	0	1	4	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0
0	0	1	0	0	2072	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	4	3	0	0	0	0	0.75	0.50	0	3	5	1	0	1	0.40	0	0	0	0
0	0	1	0	0	5014	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	2	0	0	0	0	0.67	0.67	0	2	4	1	0	1	0.50	0	0	0	0
79	0	0	0	281	1301	0	0	0	2	0	1	1	1	0	0	4	2	0	0	0	0	1	1	0	0	0	0	1	0	0	1	10	1	0	1	0.30	0	0	0	0.10
103	0	0	0	302	8876	0	0	0	2	0	1	4	1	0	3	4	2	1	0	0	0	1	1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0
31	0	0	0	142	1278	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	5	3	0.60	0.60	0.20	0	0	0	0	0
0	0	1	0	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	150	25	0.17	0.03	0.17	0	0	0	0.05	0
0	1	2	0	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	1	0	0	0	0	0.08	0.15	0	255	1	0	0.60	0.88	0	0	0	0	0
0	0	3	0	232	8153	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	5	0.20	0.20	0	0	1	0	0	30	255	1	0	0.03	0.04	0.03	0.01	0	0.01
0	0	3	0	199	420	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	30	32	0	0	0	0	11	0	0.09	255	255	1	0	0	0	0	0	0	0
0	0	3	0	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	7	0	0	0	0	1	0	0.43	8	219	1	0	0.12	0.03	0	0	0	0

Figure 2 – Sample fragment for neurons of the first layer of MLNN

1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	1	0	0
0	1	0	0
0	1	0	0
0	1	0	0
0	1	0	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1

Figure 3 – Sample fragment for neurons of the result layer of MLNN

3. Creation, training and testing the MLNN

With the help of the Fuzzy Logic Toolbox package, MatLAB created MLP configuration 41-1-25-4 (where 4 is the number of input neurons; 1 – the number of hidden layers; 25 – the total number of neurons of hidden layer; 4 – the number of resultant neurons), which is shown in Figure 4.

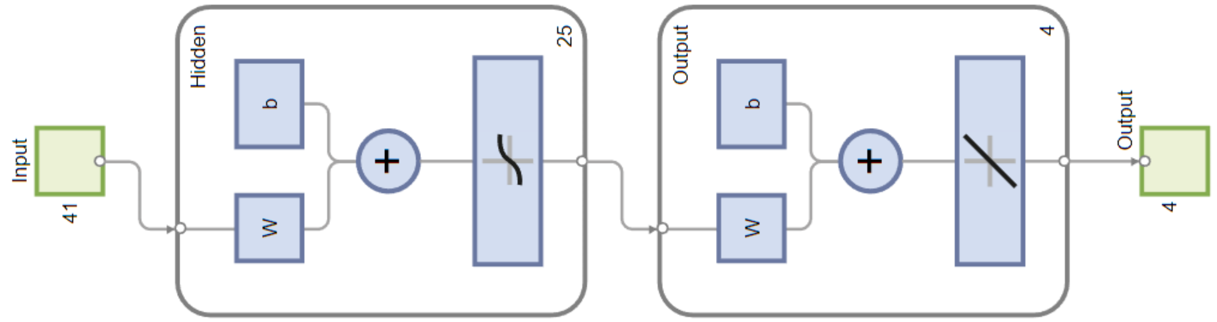

Figure 4 – Created by MLNN 41-1-25-4 in the MatLAB system

Authoring

The results of MLNN training and testing are presented in Figure 5. As can be seen from the figure, the error of MLNN was 0.08 during testing (Levenberg-Marquardt algorithm).

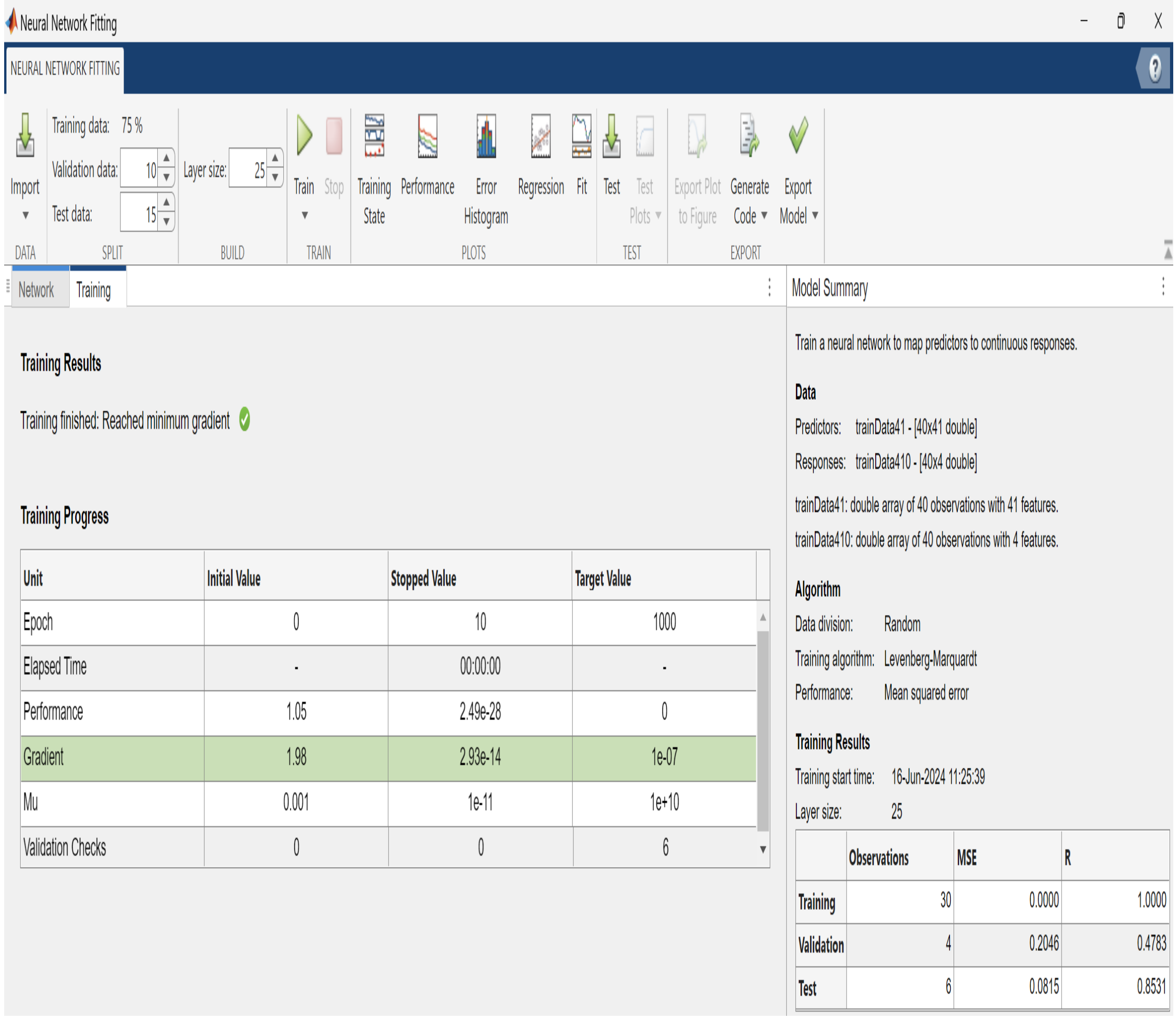

Figure 5 – MLNN training and testing (Levenberg-Marquardt algorithm)

4. Exploration of MLNN parameters

In addition, on the 41-1-X-4 configuration created by MLNN, studies of its error were carried out on samples of 40 examples using various training algorithms: Levenberg-Marquardt; Bayesian Regularization; Scaled Conjugate Gradient (Figure 6). On the configuration 41-1-25-4 created by the MLNN, the values of errors of the first and second kind are 9 % and 10 %, respectively.

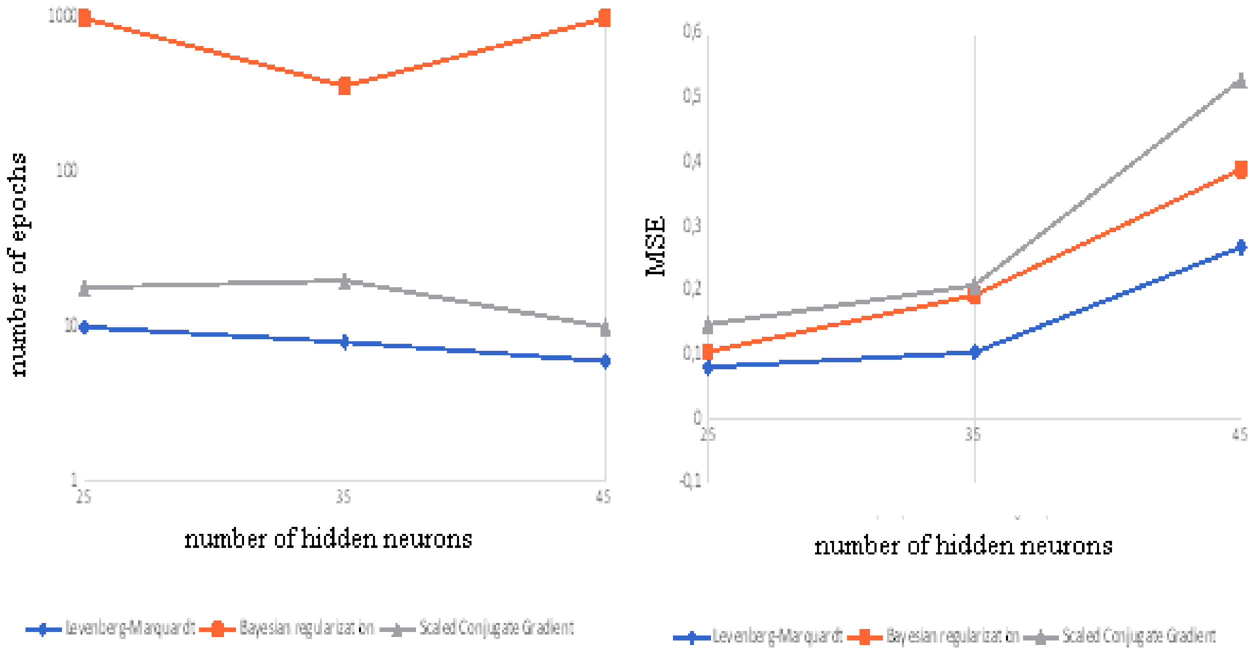

Figure 6 – MLNN number of epochs and MSE for different hidden neurons

Authoring

Conclusions

1. To detect U2R attacks, a MLNN of configuration 41-1-X-4, where 4 is the number of input neurons; 1 – the number of hidden layers; 25 – the total number of neurons of hidden layer; 4 – the number of resultant neurons (y1 – Rootkit attack, y2 – Buffer_overflow attack, y3 – Loadmodule attack, y4 – No attack).

2. In the MatLAB system, with the help of the package Neural Network Toolbox, an MLNN configuration 41-1-X-4 was created, a hyperbolic tangent was taken as a function for activating the hidden layer, of the resulting layer is a linear function. It was determined that the MSE was approximately 0.08 according to the Levenberg-Marquardt algorithm on a sample of 40 examples, the data for which were taken from the NSL-KDD database (10 examples for each case).

3. On the 41-1-X-4 configuration created by MLNN, MSE and the number of epochs of training at different numbers of hidden neurons were studied using different training algorithms on samples of different lengths. The optimal parameters of MLNN have determined.

4. An assessment of the quality of detection of U2R attacks on MLNN configuration 41-1-25-4 at its optimal parameters was carried out. It is determined that errors of the first and second kind are 9 % and 10 %, respectively.

References

1. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. *University of New Brunswick | UNB*. URL: https://www.unb.ca/cic/datasets/nsl.html
2. Pakhomova V., Kuluk V. (2022). Study of the possibility of using the RBF network to detect U2R category network attacks. *ScientificWorldJournal*. Bulgaria. Issue 16. Part 1. pp. 30-35. URL: https://doi.org/10.30888/2663-5712.2022-16-01-036.
3. Pakhomova V., Mihelbei Y. (2022). Detection of attacks of the U2R category by means of the SOM on database NSL-KDD. *System Technologies*. No 5(142). pp. 18-27. URL: http://eadnurt.diit.edu.ua/jspui/handle/123456789/16940.

Анотація. *У якості методу дослідження використано багатошарову нейронну мережу конфігурації 41-1-Х-4, де 41 – кількість вхідних нейронів; 1 – кількість прихованих шарів; Х – загальна кількість прихованих нейронів; 4 – кількість результуючих нейронів, що створена за допомогою пакета Neural Network Toolbox системи MatLAB, для виявлення мережевих атак категорії U2R: y1 – атака Rootkit, y2 – атака Buffer overflow, y3 – атака Loadmodule, y4 – відсутність атаки. З використанням відкритої бази даних параметрів мережевого трафіку NSL-KDD на створеній нейронній мережі проведено дослідження її похибки та кількості епох навчання при різній кількості прихованих нейронів (25, 35 та 45) за різними алгоритмами навчання: Levenberg-Marquardt; Bayesian Regularization; Scaled Conjugate Gradient. Досліджено, що найменше значення похибки нейронної мережі склало з використанням гіперболічного тангенсу у якості функції активації прихованого шару за алгоритмом навчання Levenberg-Marquardt, при цьому достатньо мати 25 прихованих нейронів. Проведено оцінювання якості виявлення мережевих атак категорії U2R на нейромережі конфігурації 41-1-25-4 при її оптимальних параметрах. Визначено, що помилка першого та другого роду складає 9 % та 10 % відповідно.*

Ключові слова: *U2R, трафік, NSL-KDD, MLNN, гіперболічний тангенс, похибка, помилка першого роду, помилка другого роду.*