NEURAL NETWORK MODEL OF COUNTERING NETWORK CYBER ATTACKS USING EXPERT KNOWLEDGE

IDEYAT BAPIYEV, GAUKHAR KAMALOVA, FARIDA YERMUKHAMBETOVA, AIMGUL KHAIRULLINA, AKMARAL KASSYMOVA

West Kazakhstan Agrarian Technical University named after Zhangir Khan, Uralsk, West-Kazakhstan Region, Kazakhstan

E-mail: ideyat.bapiyev@gmail.com

ABSTRACT

Research in the field of countering cyberattacks on network resources of information systems has shown that most modern neural network models are focused on learning using statistical data. Such models are not sufficiently adapted to recognize new types of network cyberattacks. To eliminate this drawback, it was proposed to form a training sample using expert knowledge presented in the form of production rules. It was determined that among the classical types of neural network models, the most suitable for such training is a probabilistic neural network. On the basis of this network, an original neural network model was created, its structure and software were developed. The use of the developed model makes it possible to increase the recognition efficiency and expand many types of network attacks, the characteristics of which are not presented in statistical data. Another important advantage of the developed model is the information content of the output signal, which is sufficient for flexible setting of protective measures.

Keywords: *Neural Network Model, Recognition Of Cyber Attacks, Model, Production Rule, Expert Knowledge, Probabilistic Neural Network.*

1. INTRODUCTION

In modern conditions, one of the most urgent problems in the field of information security is to increase the effectiveness of systems of recognizing cyber attacks on the network resources of information systems [1, 2]. At the same time, one of the most promising areas of increasing recognition efficiency is the use of intelligent methods based on the use of neural network models [3]. Perspectivity of this area is confirmed by the well-known successful cases of using neural network models in the recognition systems of cyber attacks (Cisco products) and by a large number of relevant theoretical and practical studies, the analysis of which has been deep enough carried out in [4, 5]. However, a large number of incorrect triggers, a significant development period, instability of training, exactingness to the volume and quality of the training sample, insufficient adaptation to many features of modern information systems, connected primarily with methodological drawbacks, significantly reduce their practical value. Thus, it is necessary to quickly solve the problem caused on the one hand by the necessity and the prospect of using existing neural network tools of recognizing cyber attacks on network resources, and on the other hand by the imperfection of methods of developing and using such means.

The aim of the work is to develop effective models, methods and means of countering cyberattacks adapted to the operating conditions and capable of promptly responding to new types of network cyberattacks. The main criterion for the selection of literature is the work associated with the use of the apparatus of artificial neural networks. The hypothesis of the study is the assumption that a neural network model of countering cyberattacks, based on the possibility of training with the help of expert data, will allow to respond quickly to the types of network cyberattacks.

In general, according to [4, 5], the concept of cyber attack is understood as the realization in cyberspace of threats to the security of its components (namely, confidentiality, integrity and accessibility), taking into account their vulnerabilities.

In the foreground of the subject-matter of this article, those cyber attacks are considered that are implemented only using network traffic parameters and take into account the vulnerabilities of only network resources of information systems. The purpose of such cyber attacks is to violate the correctness of the network functional characteristics of the attacked resources. Typical examples of such cyber attacks are DOS and DDOS attacks.

Analysis of scientific and practical works, devoted to improvement of Cyber attack detection tools allows to state that in these systems the NS are used to recognize cyber attacks on the basis of generalization of statistical data are displayed in the training set [3-5]. In addition, it is possible to formulate the conclusion that most of the relevant studies are aimed at adapting the architecture of neural network models to the conditions of practical task.

Thus, in [6], a method of optimizing the type of the NM architecture was developed, depending on the specifics of the posed recognition problem. In [1, 7], the peculiarities of the NNM application for recognition of selection attempts of the password and recognition of DDOS attacks. The work [8] is devoted to solving the problem of improving the structure and learning algorithm of the multilayer perceptron intended for use in Cyber attack detection tools. Another common area of research is the application development methods in little tested types of neural network models in the Cyber attack detection tools. For example, work [3] is devoted to recognition means based on cyber neuron, and in [9] it is proposed to use the Kohonen map, functioning in accordance with the principles of artificial immune systems.

At the same time, critics [4, 5] are justified on the fact that the use of classical neural network models is inherently one of the ways to process statistical information, as well as many types of cyber attacks, represent a set of non-standard operations, the characteristics of which are practically not reflected in the presented statistics-the training sample. Therefore, it is possible to recognize a new kind of cyberattack only after it was implemented, and its parameters were fixed and displayed in the training sample.Thus, a significant drawback of most modern neural networked Cyber attack detection tools is not sufficient adaptation to new types of cyber attacks.

In our opinion, a promising way to fix this drawback is the use of the method developed in [4-6] of training the neural network models of Probabilistic neural network type with the help of expert knowledge written in the form of production rules.

However, the neural network models developed in [4-6] has insufficient information content of the output signal, which, in the authors' opinion, makes it difficult to use it for recognizing network cyber attacks.

Thus, **the purpose** of this study is to develop a neural network model that can be trained with the help of production rules and is designed for use in recognition systems of cyber attacks on the network resources of information systems.

2. THE CONCEPT OF APPLICATION OF PRODUCTION RULES TO GENERATE THE INPUT AND OUTPUT PARAMETERS OF THE TRAINING EXAMPLES OF A NEURAL NETWORK

In the course of the research it was planned to develop a neural network system for countering network cyber attacks. The level of development meets modern requirements because a whole complex of fairly accurate classical and modern research methods was used (methods of the theory of digital signal processing, neural networks, expert analysis, mathematical statistics and optimization, etc.). The level of research is confirmed by approbation and scientific publications on the topic of research.

All the results obtained in the work are either based on known theoretical information, or proven and supported by the use of modern scientific methods of analysis and research.

The experimental setup is a hardware and software complex designed for experimental research of the developed models and methods, as well as a neural network system for recognizing network cyber attacks created on their basis.

Under the production rule, we understand the rule given by an expression of the form:

$$Ifcondition = \text{true/false} \to (\text{output}) \quad (1)$$

From the point of view of the NN theory, such a rule corresponds to basic training example of the form:

$$X \to Y, \quad (2)$$

where X – value of the input signal, Y is the expected output signal of the NN.

In this training example, one input parameter corresponds to one expected output parameter.

Note that for most practical tasks such training examples are too simplistic. More typical examples are:

$$\{X_1,...X_K\} \rightarrow Y \ , \qquad (3)$$

$$\{X_1,..X_K\} \rightarrow \{Y_1,...Y_M\} \ , \qquad (4)$$

where X_k is the value of the k-th input parameter, Y_m is the value of the m-th output parameter, K is the number of input parameters, M is the number of output parameters.

From the standpoint of the methodology of processing expert knowledge, a training example of the form (3) corresponds to a set of product rules given by the following expression:

$$If \ x_1 = X_1 \wedge ...x_K = X_K \rightarrow y = Y \ . \qquad (5)$$

where $x_1,...x_K$ – set of input parameters, y – output parameter, $X_1,...X_K, Y$ – specified constants.

A set of product rules corresponding to a training example of the form (4) can be specified by the following expression:

$$If \ x_1 = X_1 \wedge ...x_K = X_K \rightarrow y_1 = Y_1 \wedge ...y_M = Y_M \qquad (6)$$

where $y_1,...y_M$ – set of output parameters, $Y_1,...Y_M$ – specified constants.

The application of production rules for the formation of training examples of the form (3, 4) will consist in the fact that expressions (5, 6) are determined expertly in which the values of the input and output parameters correspond to a certain class. It should also be noted that in many cases not only point, but also interval values of input parameters are estimated in production rules.

3. DEVELOPMENT OF THE BASIC NEURAL NETWORK MODEL

As the results of [4-6] show, the process of formation of production rules for the recognition of cyber attacks is associated with significant difficulties. Difficulties arise due to lack of qualified experts, the complexity of the described processes of recognition, lack of the necessary data. In consequence of these difficulties, it can be assumed that the number of such expert rules will be quite small. Therefore, the training examples created on the basis of them can be used only in the NNM, which is able to learn by the method "with trainer" on a limited training set. Among the classical NNMs, only the Probabilistic neural network has this capability [8].

In this network, the classification of unknown patterns is implemented on the basis of estimates of their similarity with the training examples using the Bayes method. An unknown pattern belongs to a class which of distribution density in the region of this example is the largest. To estimate the distribution density in the area of a particular case study, the Gaussian function is used with the center at the point to which this example corresponds.

The classical Probabilistic neural network consists of four layers of neurons - input (LN_{in}), patterns (LN_o), summation and comparison (LN_{cool}), which is also an output layer. The number of neurons in the input layer equals the number of controlled parameters, the analysis of which allows to classify an unknown pattern. The number of neurons in the layer of patterns is equal to the number of training examples and the number of neurons in the summation layer is equal to the number of classes that must be recognized. The neurons of the input layer and the pattern layer form a full-meshed structure. Each neuron of the pattern layer is associated only with the neuron of the summation layer which corresponds to the class of the given pattern.

For connections entering the pattern layer neuron, the weighting factors are set the same as the normalized components of the corresponding training example.

The weight coefficients of the connections entering into the neurons of the summation layer and into the output layer neuron are equal to 1. Thus, the structure and weight coefficients of communications of Probabilistic neural network are directly determined by the training data.

The structure of the Probabilistic neural network, which is designed to classify the two Ka and Nm classes (states) based on the analysis of N controlled parameters is shown in Figure 1. In this network, the neurons of the pattern layer with numbers from 1Ka to L_{Ka} correspond to training examples that correspond to the Ka class, and neurons with numbers from 1_{Nm} to M_{Nm} correspond to the class Nm.

The output signal of the jth neuron of the pattern layer (θ_j^o) is calculated as follows:

$$\theta_j^o = \sum_{i=1}^{N} exp\left(\frac{-(w_{i,j} - x_i)^2}{2\sigma^2}\right), \qquad (7)$$

where x_i – i- i-th component of the unknown pattern, $w_{i,j}$ is the weighting coefficient of the connection between the i-th input neuron and the j-th neuron of a layer of patterns, N is the number of components of the input vector-pattern (the number of controlled parameters), σ –radius of function of Gauss.

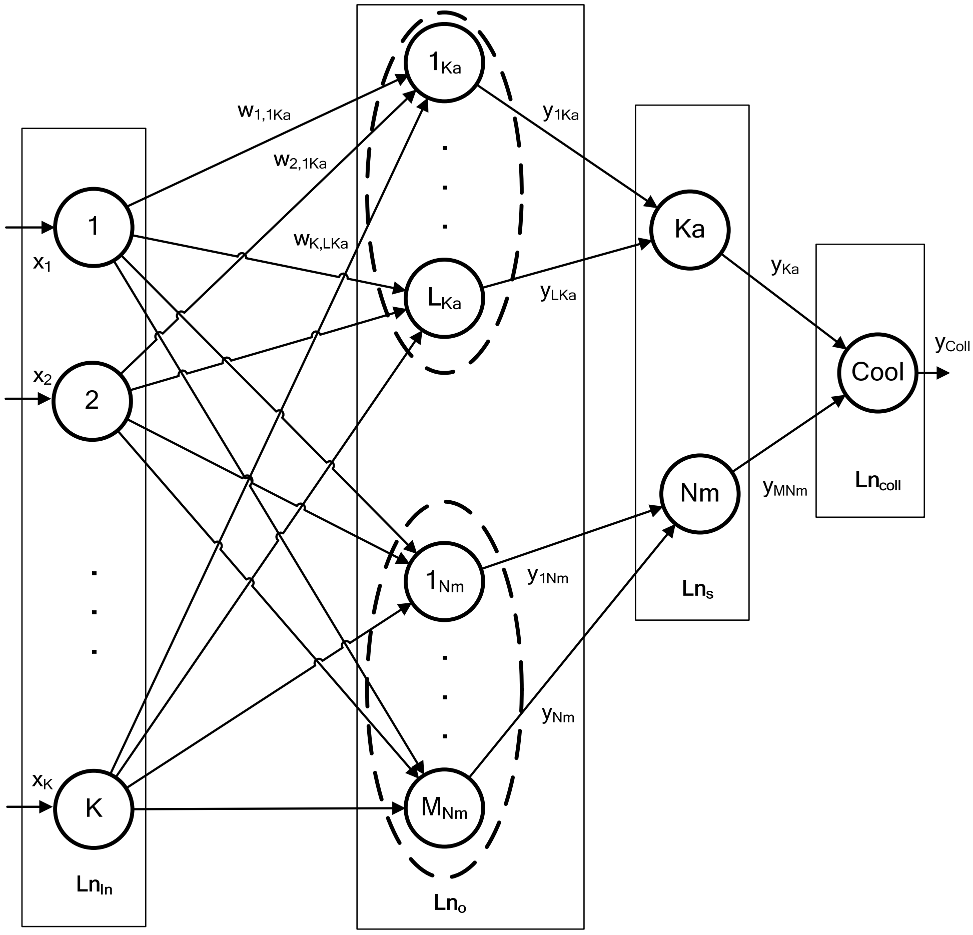


Figure 1: Structure of the Probabilistic neural network

In the neurons of the summation layer, a linear activation function is used. The output signal of the j-th neuron of this layer (θ_j^s) is calculated as follows

$$\theta_j^s = \sum_{i=1}^{N} \theta_i^o \ , \qquad (8)$$

where N is the number of neurons in the pattern layer associated with the jth neuron of the summation layer, θ_i^o is the activity of the ith neuron of the pattern layer associated with the j-th neuron of the summation layer.

The output signal of the neuron of the summation layer is proportional to the probability of assigning the input pattern to a class that corresponds to a given neuron.

The task of the output neuron (comparison neuron) is only the determination of the neuron of the summation layer with maximum activity.

In many cases, the output neuron is absent, and the neuron definition of the summation layer with maximum activity is implemented by means that are not included in PNN. Note that according to [4], in order to improve the efficiency of the calculation process of the output signal, it is advisable to represent the PNN network in a matrix form.

In this case, the matrix elements will be the weight coefficients of the connection between adjacent layers of neurons. If the connection between neurons is not provided, then it is considered that its weight coefficient is 0.

The only empirical parameter of the PNN network is the radius of the Gaussian function (σ), which is used in the expression (3) to calculate the output signal of the pattern layer neuron.

In the theoretical paper [8], it is noted that for many practical cases, $\sigma = 1$ can be accepted in the first approximation.

In order to introduce knowledge of the production rule in such NNM to classify the Ka state or the Nm state, it is necessary:

-to determine two neurons Ka and Nm in the summation layer, which correspond to given states;

-to add a new neuron to the pattern layer, which corresponds to the production rule - the training example;

-for a new neuron, to equate the weight coefficients of the incoming communications with the values of the parameters of the production rule;

- To link the new neuron with the corresponding neuron of the summation layer Ka or Nm.

For example, Figure 1 shows the weighting factors $w_{1,1_{KA}}$, $w_{2,1_{KA}}$, $w_{K,1_{KA}}$, due to which a study example was introduced into the PNN network-a production rule corresponding to the state of Ka.

4. NEURAL NETWORK MODEL OF THE RECOGNITION OF CYBER ATTACKS ON NETWORK INFORMATION SYSTEMS RESOURCES (ISR) USING THE EXPERT KNOWLEDGE

Based on the proposed concept of application of product rules for the formation of the parameters of training examples, it is determined that in accordance with [6, 9], in the first approximation, it is possible to use the parameters of network traffic as identifying parameters of cyber attacks that are applied in production rules of the form (5, 6) .

Such parameters may include the frequency of network requests, the load of the communication line, the number of incorrect packets, the protocol by which data is transferred, the processor load, the IP address from which data is transmitted, etc.

With this assumption in mind, expression (5) is modified as follows:

$$If\ x_1 = X_1 \wedge \ldots \wedge x_K = X_K \to Y_l,\ l = 1..L\,, \quad (9)$$

where $x_1,\ldots x_K$ are the values of the parameters of network traffic, K is the number of network traffic parameters, $X_1,\ldots X_K$ – threshold values, $x_1,\ldots x_K$, l is the product rule number, L is the number of production rules.

If the KDD-99 database [5] is used as a basis for forming expert rules, the number of input parameters of the NNM will be equal to K = 41, which is equal to the number of fields of the specified database.

Note that the values of these fields correspond to the values of network traffic over the TCP protocol. It should also be noted that if the KDD-99 database is used, the result of the production rule can be either a conclusion about the normal state of the network ISR, or a conclusion about one of the types of cyber attacks listed in Table 1.

Table 1: The characteristics of the KDD-99 database

Number of the type of cyber attack	Type of cyber attack	Class of cyber attack	Number of records
1	neptune	DoS	1072017
2	smurf	DoS	2807886
3	Pod	DoS	264
4	teardrop	DoS	979
5	land	DoS	21
6	back	DoS	2203
7	buffer_overflow	U2R	33
8	loadmodule	U2R	9
9	perl	U2R	3
10	rootkit	U2R	10
11	guess_passwd	R2L	53
12	ftp_write	R2L	8
13	imap	R2L	12
14	phf	R2L	4
15	multihop	R2L	7
16	warezmaster	R2L	20
17	warezclient	R2L	1020
18	spy	R2L	2
19	portsweep	Probe	10413
20	ipsweep	Probe	12481
21	satan	Probe	15892
22	nmap	Probe	2316

Considering the data of Table 1 the neural network models should classify 23 states of network ISR (22 types of cyber attacks plus one normal state).

In the case of the production rules definition by expressions (5, 6), the expert task is to determine the threshold values of the network traffic parameters (X_1, X_2,…X_K) for each of the classified network information system resources states.

At the same time, as already mentioned above, the negative feature of the Probabilistic neural network is the low informational content of the output signal. In this neural network models, the value of the output signal indicates only that the probability of which event is greater - one of the cyber attacks or normal. This does not allow the use of the established threshold values of the probability of a cyber attack/normal state or the threshold value of the difference between these probabilities in the ISF. Thus, in the ISF, the recognition unit of which functions on the basis of the classical PNN network, it is impossible to implement the generally accepted rules for the implementation of protective measures of the type:

$$If\ \Theta_{Ka} > \Delta_{Ka} \rightarrow Z \qquad (10)$$

$$If\ \Theta_{Nm} < \Delta_{Nm} \rightarrow Z \qquad (11)$$

$$If\ (\Theta_{Ka} - \Theta_{Nm}) > \Delta_{\Delta} \rightarrow Z \qquad (12)$$

$$If\ \Omega_i > \Delta_i \rightarrow Z \qquad (13)$$

where Θ_{Ka}, Θ_{Nm} – estimated probability of cyber attacks implementation, Θ_{Nm}– estimated probability of normal state, Z – protective measures, Ω_i – estimated probability of cyber attacks of i type, $\Delta_{Ka}, \Delta_{Nm}, \Delta_{\Delta}, \Delta_i$ – preset threshold values.

Beyond that the nonsufficient information content of the neural network models's output signal does not allow the ISF to respond flexibly to various combinations of network traffic parameters, and also does not allow to record the security states parameters which have probability different from the maximum probable state. Possibility to register such parameters could allow the creation of a database designed to refine the rules for recognizing known types of cyber attacks and the formation of rules for the recognition of unknown cyber attacks.

To correct these shortcomings, based on [6] it has been defined that the output information of the Probabilistic neural network, designed for the recognition of cyber attacks on network information system resources, should be supplemented by estimates:

-Probabilities of every known types of cyber attacks.

-Probabilities of the most probable kind of cyberattack.

-Probability of all types of cyber attacks.

-Probabilities of normal state.

For this purpose, the Probabilistic neural network model must be supplemented by appropriate output connections, as well as an additional sphere of neurons designed to calculate the probabilities of known cyberattack types.

The structure of the modified Probabilistic neural network intended for recognizing the state of network ISR on the basis of production rules of the form (6) is shown in Figure 2. In order to distinguish such network from the classical Probabilistic neural network, it was called a SPNN network.

Structurally, the Modified probabilistic neural network consists of 5 neural layers. This is an input layer (Ln_{in}), a pattern layer (Ln_o), the first summation layer (Ln_{s1}), the second summation layer (Ln_{s2}), and a comparison layer (Ln_{coll}).

As in the classical Probabilistic neural network, the input layer neurons only transfer data from the external environment to the neurons of the image layer. The pattern layer is intended for memorizing the parameters of training examples for all classes that the neural network models should recognize. Respectively, the number of neurons in the pattern layer is equal to the number of training examples that memorized by neural network models. In Figure 2 neurons of the pattern layer are conditionally divided into groups that correspond to known types of cyber attacks and the normal state of the network information system resources. For example, the neurons labeled 1_{KaN}…L_{KaN} correspond to the training examples of the N-th type of cyber attacks, and the neurons labeled 1_{Nm}…M_{Nm} correspond to the training examples of the normal state. Also note that in general, the number of training examples for each type of cyberattack and for the normal state can be different.

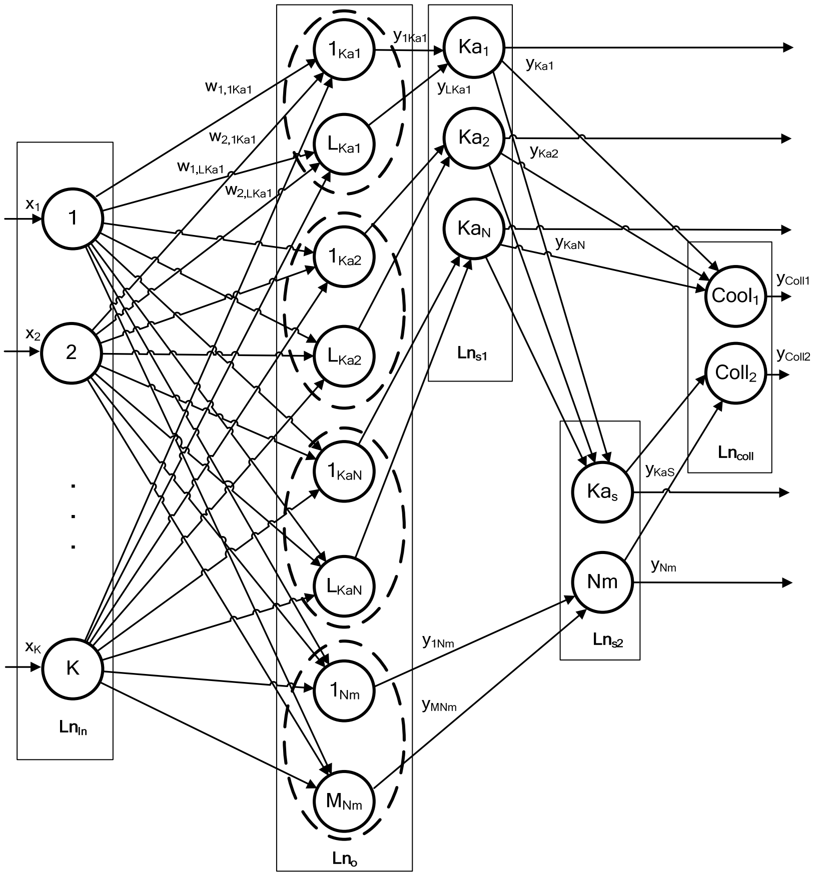


Figure 2: Structure of Modified probabilistic neural network

The task of the first summation layer is to estimate the probability of each type of cyberattack known for neural network. Therefore, the neurons of this layer are associated with the known types of cyber attacks $Ka_1, Ka_2, \ldots Ka_N$, and their number is equal to the number of these types of cyber attacks (*N*). The second summation layer is intended to determine the estimation of the total probability of all types of cyber attacks and estimated probability of network РИС's normal state. This layer contains two neurons. The neuron Ka_s is used to calculate the total probability of cyber attacks, and the neuron Nm is used to calculate the probability of the normal state. The comparison layer's task is to determine the type of the most probable cyberattack, and also the type of the most probable security state. The type of the most probable cyber attack is determined by neuron $Coll_1$, and the most probable protection state (cyberattack is implemented or normal state) is determined by neuron $Coll_2$.

The structure between the Ln_{in} and Ln_o layers is fully connected. In this case, each neuron of the input layer is connected by a direct connection to each neuron of the pattern layer. Weight coefficients of these connections are equal to the components of the training examples:

$$w_{1,l} = X_1, w_{2,l} = X_2, \ldots w_{K,l} = X_K \,, \qquad (14)$$

where l – number of the pattern sphere neuron, which corresponds to the number of the production rule of the type (6),

K – number of input parameters, $X_1, X_2, \ldots X_K$ – components of the l -th production rule of the type (2.111).

Figure 2 shows the weighting coefficients of connections between the first input neuron and $1_{Ka1} \ldots L_{Ka1}$, $1_{Ka2} \ldots L_{Ka2}$ neurons of the pattern layer. Note that the weighting coefficients of all other connections in the SPNN network are equal to 1. In neurons of the pattern layer the Gaussian function is used as activation function. The output signal of undefined l-th of the neurons of the pattern layer is calculated as follows:

$$y_l = \sum_{k=1}^{K} \exp\left(-(w_{k_i,l} - x_k)^2 / 2\sigma^2\right), \qquad (15)$$

where $w_{k_i,l}$ – the weight coefficient of the connection between the k-th neuron of the input layer and the l-th neuron of the pattern layer, x_k – quantity of the k-th input signal, K – number of input parameters, σ – radius of the Gaussian function.

Note that for the problem of network cyber attacks recognition the recommended radius of the Gaussian function is $\sigma \in [0{,}3...{,}0{,}7]$. The connection structure between the pattern sphere and the first summation sphere has a peculiarity. Only those pattern sphere neurons which correspond to the parameters of the cyberattack type with which this neuron is associated are connected with the neuron of the first summation sphere. For example, as shown in Figure 2, the neurons $1_{Ka1}...L_{Ka1}$ connect only to neuron Ka_1. A linear activation function is used in the neurons of the summation layers. The output signal for an undefined n-th neuron of the first summation layer is calculated as follows:

$$y_{Ka_n} = \frac{\sum_{l=1}^{L_{Ka_1}} y_{l_{Ka_1}}}{L_{Ka_1}}, \qquad (16)$$

where L_{Ka_1} – number of pattern layer neurons connected with the n-th neuron of the first summation layer, $y_{l_{Ka_1}}$ – output signal of the l-th pattern layer neuron connected with the n-th neuron of the first summation layer.

It is possible to estimate probability of each known type of cyberattack $Ka_1,...Ka_N$ by the quantity of the output signals $y_{Ka_1},...y_{Ka_N}$. These signals form part of the Modified probabilistic neural networks output information. Besides they are transferred to both the neuron $Cool_1$ of the comparison layer and into the neuron Ka_s of the second summation sphere. The output signal of this neuron is calculated as follows:

$$y_{Ka_s} = \sum_{n=1}^{N} y_{Ka_n}, \qquad (17)$$

where N – number of known cyber attacks.

The value y_{Ka_s} that is proportional to the total probability of all types of cyber attacks is also part of the SPNN network's output information. In addition, y_{Ka_s} is transferred to the neuron $Cool_2$ of the comparison layer for further processing.

The output signal of the neuron Nm of the second summation layer is a probability estimate of the network ISR's normal state. This neuron is connected with only those neurons of the pattern layer that are connected with the teaching examples of the normal state.

Value of the neuron Nm's output signal is calculated as follows:

$$y_{Nm} = \frac{\sum_{m=1}^{M_{Nm}} y_{Nm_m}}{M_{Nm}}, \qquad (18)$$

where M_{Nm} – number of pattern sphere neurons that correspond to the training examples of the normal state, y_{Nm_m} – output signal of m-th neuron of the patterns sphere, that corresponds to the m-th example of the normal state.

As y_{Ka_s} signal y_{Nm} that is part of the network output information is transferred to $Cool_2$ neuron.

In the output neurons $Cool_1$ and $Cool_2$ of the comparison layer, the most probable type of cyberattack is determined and the most probable security state is defined - the cyberattack is implemented/normal state. In Figure 2 the corresponding output signals are designated as y_{Coll_1} and y_{Coll_2}.

It is necessary to perform five steps in order to introduce the l -th production rule of type (6) into the SPNN network, which determines the presence of a cyberattack of a new Ka_J type:

1. to introduce a new neuron l_{Kan} into the pattern layer
2. to connect neuron l_{Kan} to the input neurons and, in accordance with the expression (14), to set the weight coefficients of the input connections for this neuron
3. to introduce a new Ka_J neuron in the first summation layer
4. to connect neuron Ka_J to neuron l_{Kan}.
5. to connect neuron Ka_J to neuron Ka_S, which calculates the total probability of all known type's cyber attacks.

In order to introduce the SPNN m-th production rule type (6), defining the normal state of network ISR, it is necessary:

1. to introduce a new neuron m_{Nm} into the pattern layer

2. to connect the neuron m_{Nm} to the input neurons and using the expression (14), to set weight coefficients of the input connections for it equal to the corresponding components of the production rule.

3. to connect neuron m_{Nm} to neuron Nm, which calculates the total probability of network ISR's normal state.

In the mode of network ISR's security status recognition, the SPNN network functions as follows:

1. Parameter vector $\{x\}_K$ that characterizes unknown state of the network ISR security is given to the NNM input.

3. Using the expression (2.117), the output signal of each pattern sphere neurons Ka$_1$, Ka$_2$,…Ka$_N$ is calculated.

4. Probabilities of each known types of cyber attacks are estimated. For this purpose, using the expression (16), the output signal of each neuron of the first summation sphere is calculated.

5. Integral probability of all types of cyber attacks is estimated. For this end, using the expression (17), the output signal of the neuron Ka$_s$ belonging to the second summation sphere is calculated.

6. Integral probability of the network ISR's normal state is estimated. In this regard, using the expression (18), the output signal of the neuron Nm belonging to the second summation sphere is calculated.

7. The most probable cyberattack is defined. To do this, using the neuron Cool$_1$, the neuron of the first summation layer of the output signal with the largest value is calculated. The output signal $y_{Coll1} = k$, where k is the neuron number of the first summation layer with the largest output signal. The cyberattack associated with this neuron is considered the most probable. The most probable security state is determined.

To do this, using the Cool$_2$ neuron, the values of the output signals of neurons Ka$_s$ and Nm are compared. If $y_{Ka_s} \geq y_{Nm}$, then $y_{Coll1} = Ka_s$ it is considered that network ISR is exposed to a cyberattack. In this case $y_{Coll1} = k$. In the opposite case $y_{Coll1} = Nm$, it is considered that the state of network ISR is normal.

5. EXPERIMENTAL RESEARCH

To illustrate the main features of the proposed NNM, an example of its application for security state parameters definition of a certain network resource is considered.

Problem statement.

- The security state can be estimated by value of two monitored security parameters x_1 and x_2.

- The current value of the monitored security settings $x_1 = 14$, $x_2 = 28$.

- the network resource can be exposed to cyber attacks of two kinds of Ka$_1$ and Ka$_2$, for the recognition of which it is possible to apply four production rules of the form (2.115).

Rule 1: $\textit{If } x_1 = 5 \wedge x_2 = 7 \to Ka_1$

Rule 2: $\textit{If } x_1 = 12 \wedge x_2 = 4 \to Ka_1$

Rule 3: $\textit{If } x_1 = 15 \wedge x_2 = 50 \to Ka_2$

Rule 4: $\textit{If } x_1 = 18 \wedge x_2 = 56 \to Ka_2$

- To determine the normal state of the network resource, two production rules of the form (2.111) are possible.

Rule 5: $\textit{If } x_1 = 37 \wedge x_2 = 15 \to Nm$

Rule 6: $\textit{If } x_1 = 20 \wedge x_2 = 24 \to Nm$

Solution.

1. In accordance with the above algorithms of entering production rules into the SPNN network, the structure of the NNM is constructed and the weight coefficients are determined. The structure is shown in Figure 3, and weight coefficients are presented in Table 2. Accepted $\sigma = 0{,}5$.

2. In accordance with expressions (15-18), the output parameters of the SPNN network have been calculated when the current values of the monitored security parameters are given to its input. The result is:

$$y_{Ka_1} = 1{,}68 \times 10^{-4}, \qquad y_{Ka_2} = 6{,}77 \times 10^{-2},$$
$$y_{Ka_S} = 6{,}78 \times 10^{-2}, \quad y_{Ka_1} = 6{,}33 \times 10^{-15}, \quad y_{Coll_1} = Ka_2,$$
$$y_{Coll_2} = Ka.$$

3. As a result of the analysis of the output parameters, it can be determined that at the current time the network ISR is exposed to the cyber attack of the form Ka_2.

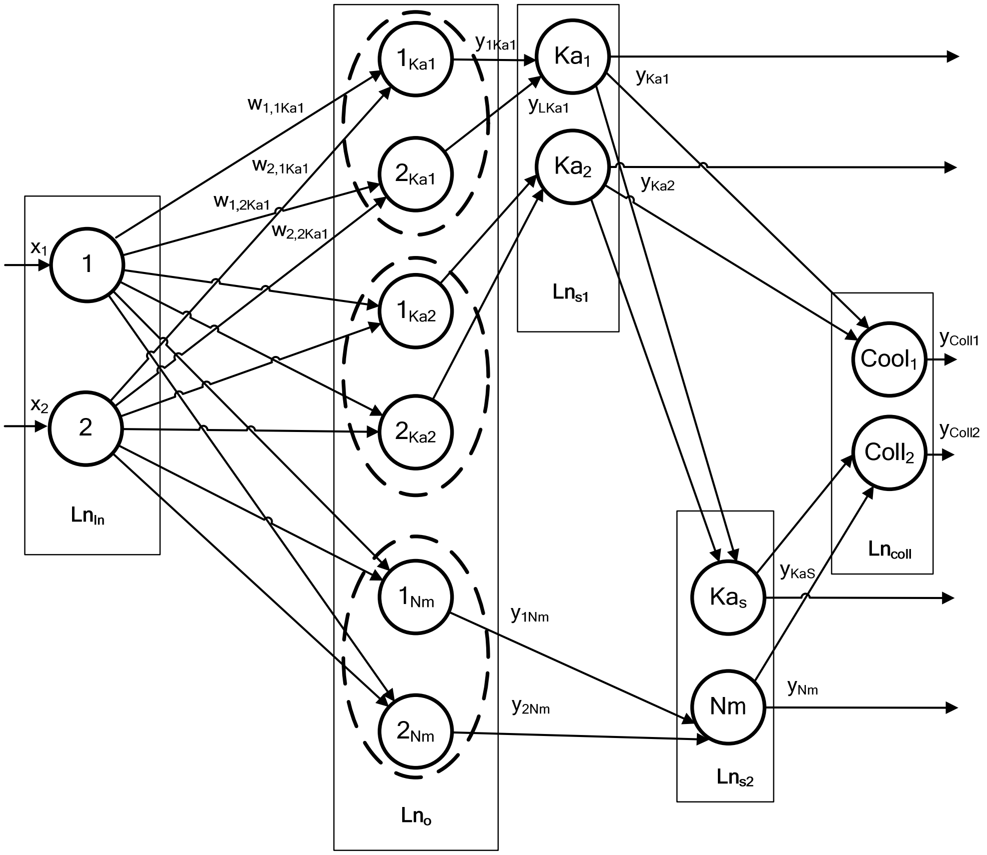


Figure 3: The structure of the Modified probabilistic neural network, adapted to the conditions of the task

Table 2: The values of weight coefficients

Production Rule Number	Designation of the pattern sphere neuron	Designation of the input neuron	
		x_1	x_2
1	1_{Ka1}	5	7
2	2_{Ka1}	12	4
3	1_{Ka2}	15	50
4	2_{Ka2}	18	56
5	1_{Nm}	37	15
6	1_{Nm}	20	24

6. DIRECTIONS FOR FUTURE RESEARCH

Despite the possibility of presenting expert knowledge, the main disadvantage of the basic model is the low ability to generalize the training information, which hinders the widespread application of the modified SPNN network. Note that the NNM's ability of generalization is generally accepted to estimate the ratio of the number of synaptic connections to the number of training examples, which it can memorize without error or with a certain error. For a Probabilistic neural network, one pattern sphere neuron with a number of synaptic connections corresponds to one training example, which exceeds the number of input parameters per unit. At the same time, 10-100 training examples correlate to a multilayer perceptron with one output neuron with the same number of synaptic connections [5, 8, 10]. Accordingly, when recognizing network cyber attacks, the generalizing capabilities of a multilayer perceptron are 10-100 higher than in a Probabilistic neural network. However, PNN and multi-layer perceptron have very similar structural schemes and belong to the same class of NNM with signal feedforward. In addition, the analysis [5, 10] indicates that it is possible to create a multi-layer perceptron, base of which is the Probabilistic neural network, using the constructive algorithms. Therefore, it is promising to develop a method of applying product rules for training a multi-layer perceptron designed to recognize network cyber attacks. Another important area of improving the proposed neural network model should be its adaptation to the use of expert knowledge about network cyber attacks submitted with the help of the fuzzy logic apparatus.

7. CONCLUSION

The kddNeural program is written in the C # programming language.

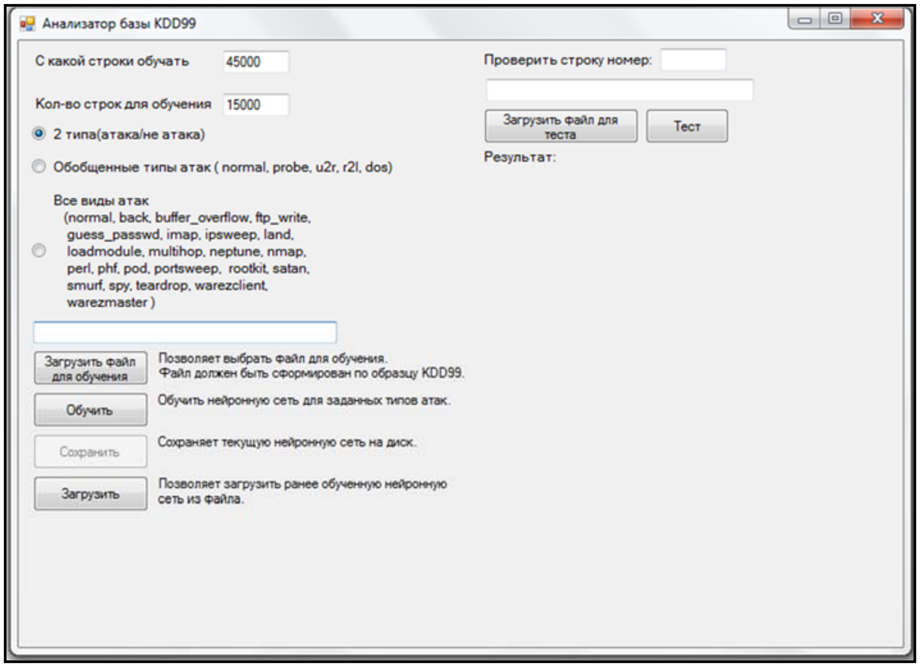
kddNeural allows you to:

– Classify network requests by type - normal or cyberattack.

– Recognize classes of network requests - normal, u2r, probe, r2l, dos.

– Recognize the following types of network cyber attacks - back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, normal, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster.

The main window of the program is shown in Figure 4.


Анализатор базы KDD99

С какой строки обучать 45000

Кол-во строк для обучения 15000

● 2 типа(атака/не атака)

○ Обобщенные типы атак (normal, probe, u2r, r2l, dos)

○ Все виды атак
(normal, back, buffer_overflow, ftp_write,
guess_passwd, imap, ipsweep, land,
loadmodule, multihop, neptune, nmap,
perl, phf, pod, portsweep, rootkit, satan,
smurf, spy, teardrop, warezclient,
warezmaster)

Загрузить файл для обучения – Позволяет выбрать файл для обучения. Файл должен быть сформирован по образцу KDD99.

Обучить – Обучить нейронную сеть для заданных типов атак.

Сохранить – Сохраняет текущую нейронную сеть на диск.

Загрузить – Позволяет загрузить ранее обученную нейронную сеть из файла.

Проверить строку номер:

Результат:

Figure 4: The Main Window Of The Kddneural Program

In accordance with the architecture of a probabilistic neural network, the kddNeural functionality provides two modes of operation: training and recognition. There is also a service option for saving and later using the parameters of the trained neural network model. Also, in order to increase the flexibility of conducting experiments, the kddNeural program allows you to train a neural network model on a certain part of a file containing a training set.

The result of the operation of kddNeural in the recognition mode are the probabilities of the analyzed network request belonging to each of the predefined classes.

As a result of the research, the neural network model designed to detect cyber attacks on the network resources of information systems has been improved, in contrast to the known ones, allows to carry out training through the use of expert knowledge in the form of production rules. This allows to increase recognition efficiency and expand many types of network attacks, the characteristics of which are not presented in statistical data. Another important difference of the developed model is the informative value of the output signal, sufficient for the flexible setting of protective measures.

The results of scientific research can be used in the development of new and modernization of existing means of countering cyber attacks on the network resources of information systems of both general purpose and on the network resources of critical infrastructure facilities. This will increase the effectiveness of countermeasures due to more accurate recognition of the facts of the implementation of cyberattacks. Requirements for the cyber attack recognition system, the characteristics of the software and hardware of the cyber attack recognition system, and the resources allocated for its development can be used as input data for a specific application. In addition, the results obtained can be used to train specialists who are engaged in the development and operation of means of countering network cyber attacks.

REFRENCES:

[1] E.S. Abramov, “Development and analysis of attacks detection systems methods' design”, Thesis for degree of Cand. Tech. Sciences: 05.13.19, State Radio Engineering University, Taganrog, 2005, p. 199.

[2] N. Ryabchuk, N. Grishko, V. Grishko, A. Rudenko, V. Petryk, I. Bapiyev, and S. Fedushko, “Artificial intelligence technologies using in social engineering attacks”, in *CEUR Workshop Proceedings*, Vol. 2654, 2020, pp. 546-555.

[3] A.V. Artemenko, and V.A. Golovko, “Analysis of neural network methods of computer virus detection”, in *Materials of section sessions. Youth Innovation Forum "INTRI" – 2010*, State Educational Institution "BelISA", Minsk, Belarus, 2010, p. 239.

[4] A.K. Bolshev, “Algorithms of conversion and classification of traffic for intrusion detection in computer networks”, Autoabstract thesis for degree of Cand. Tech. Sciences: spec. 05.13.19: Methods and systems of information security, information security, St. Petersburg. State Electrotechnical University, St. Petersburg, 2011, p. 36.

[5] D.Yu. Gamayunov, “Detection of computer attacks on the basis of the analysis of the network objects' behaviour”, Autoabstract thesis for degree of Cand. Tech. Sciences: speciality 05.13.11: mathematical and software support of computers, complexes and computer networks, Moscow state University named after M.V. Lomonosov, Moscow, 2007, p. 11.

[6] A. Korchenko, I. Tereykovsky, N. Karpinsky, and S. Tynimbaev, *Neural network models, methods and tools for assessing the security parameters of Internet-oriented information systems:* monograph. Kiev, Ukraine: TOV "Our Format", 2016, 275 p.

[7] B. Aitchanov, and I.M. Bapiyev, “Razrabotka protsedury opredeleniya ozhidayemogo vykhodnogo signala neyrosetevoy modeli raspoznavaniya kiberatak [Development of a procedure for determining the expected output signal of a neural network model for recognizing cyber attacks]”, *International Journal of Applied and Fundamental Research*, Vol. 5, 2017, pp. 8-11. DOI 10.17513/mjpfi.11532

[8] Yu.G. Yemelyanova, A.A. Talalaev, I.P. Tishchenko, and V.P. Fralenko, “Neural network technology of detection of network attacks to information resources”, Program Systems: Theory and Applications, Vol. 3, No. 7, 2011, pp. 3-15.

[9] B. Aitchanov, A. Korchenko, I. Tereykovskiy, and I. Bapiyev, “Perspectives for using classical neural network models and methods of counteracting attacks on network resources of information systems”, *News of the National Academy of Sciences of the Republic of Kazakhstan, Series of Geology and Technical Sciences*, Vol. 5, No. 425, 2017, pp. 202-212.

[10] I.M. Bapiyev, B.H. Aitchanov, I.A. Tereikovskyi, L.A. Tereikovska, and A.A. Korchenko, “Deep neural networks in cyber attack detection systems”, *International Journal of Civil Engineering and Technology*, Vol. 8, |No. 11, 2017, pp. 1086-1092.