computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Journal
of Computing

NEURAL NETWORK APPROACHES FOR INTRUSION DETECTION AND RECOGNITION

Vladimir Golovko, Leanid Vaitsekhovich

Brest State Technical University, Moskovskaja str. 267, 224017 Brest, Belarus
gva@bstu.by, vspika@rambler.ru

Abstract: *Most current Intrusion Detection Systems (IDS) examine all data features to detect intrusion. Also existing intrusion detection approaches have some limitations, namely impossibility to process large number of audit data for real-time operation, low detection and recognition accuracy. To overcome these limitations, we apply modular neural network models to detect and recognize attacks in computer networks. It is based on combination of principal component analysis (PCA) neural networks and multilayer perceptrons (MLP). PCA networks are employed for important data extraction and to reduce high dimensional data vectors. We present two PCA neural networks for feature extraction: linear PCA (LPCA) and nonlinear PCA (NPCA). MLP is employed to detect and recognize attacks using feature-extracted data instead of original data. The proposed approaches are tested using KDD-99 dataset. The experimental results demonstrate that the designed models are promising in terms of accuracy and computational time for real world intrusion detection.*

Keywords: *Neural networks, computer security, network attack, intrusion detection, principal component analysis, multilayer perceptron.*

1. INTRODUCTION

At present time one of the form of world space globalization is cyber space globalization because of increasing number of computers connected to the Internet. The rapid expansion of network-based computer systems has changed the computing world in the last years.

As a result the number of attacks and criminals concerning computer networks are increasing. Therefore the security of computer networks becomes more and more important.

The goal of Intrusion Detection Systems (IDS) is to protect computer networks from attacks. An IDS has been widely studied in recent years. It must perform their task in real time. There exist two main intrusion detection methods: misuse detection and anomaly detection. Misuse detection is based on the known signatures of intrusions or vulnerabilities. The main disadvantage of such an approach is it cannot detect novel or unknown attacks that were not previously defined. There are examples of misuse detection models: IDIOT [1], STAT [2] and Snort [3]. Anomaly detection defines normal behaviour and assumes, that an intrusion is any unacceptable deviation from normal behaviour. The main advantage of anomaly detection model is the ability to detect unknown attacks. There are examples of anomaly detection models: IDES [4] and EMERALD [5].

There exist the different defense approaches in order to protect the computer networks, namely, neural networks, data mining, statistical approach.

The principal component classifier is examined in [6, 7]. The data mining techniques were presented in [8, 9]. The other authors proposed a geometric framework for unsupervised anomaly detection and three algorithms: cluster, *k*-Nearest Neighbor (*k*-NN) and Support Vector Machine (SVM) [10, 11]. The different neural networks can be used for intrusion detection [12, 13]: Self Organizing Maps (SOM), MLP, Radial Basis Function (RBF) network.

The major problem of existing models is recognition of new attacks, low accuracy, detection time and system adaptability. The current anomaly detection systems are not adequate for real-time effective intrusion prevention [11]. Therefore processing a large amount of audit data in real time is very important for practical implementation of IDS.

In our previous paper [14] we proposed four variants of IDS architectures. They were based on combination linear PCA neural network (LPCA) and MLP. In this paper we extend our previous work and examine several models: LPCA and MLP, NPCA and MLP, Ensembling Network (EN). PCA network

are employed for feature extraction and for dimensionality reduction. MLP is intended to identify and recognize attacks using feature-extracted data.

The paper is organized as follows. The main stages of detection process and the data, which we use, are given in Section 2. In Section 3 the intrusion detection systems are described, based on modular neural network approach. Section 4 deals with linear and nonlinear recirculation neural networks (RNN). Section 5 describes the ensembling and MLP neural networks and rules used for their training. Section 6 presents experimental results. Finally, concluding remarks are made in the last section.

2. THE DETECTION PROCESS

The detection process using the data from network traffic is illustrated in Fig.1. It consists of three stages.

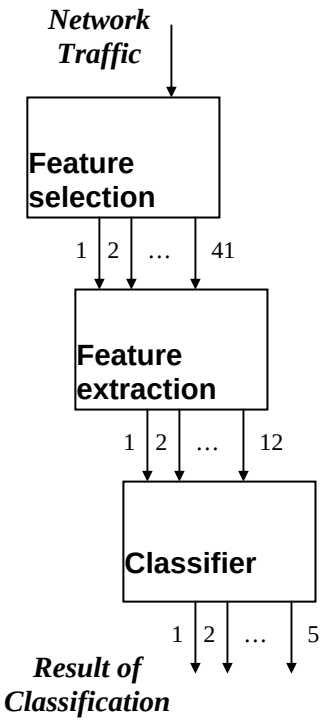

Fig. 1 – The detection process.

The first stage involves measurement of network traffic for feature selection. The special software monitor selected characteristics of the network traffic for features obtaining. In this paper we use the KDD-99 data set [15]. The data set contains approximately 5000000 connection records. Each record in the data set is a network connection pattern, which is defined as a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol.

Every record is described by 41 features and labeled either as an attack or non-attack. Every connection record consists of about 100 bytes. Among these features, 34 are numeric and 7 are symbolic. For instance, the first one is the duration of connection time, second is protocol type, and third is service name, and so on. Therefore in the first stage the features are converted into a standardized numeric representation.

The second stage involves feature extraction for important data selection and dimensionality reduction. Between the selected features exist complex relationships, which are difficult to discover. Some data may be redundant and not useful for IDS. Large amount of features can increase computation time. Therefore feature extraction is very important stage. In this paper we use linear and nonlinear PCA neural networks (RNN) for important data extraction. As a result we extract 12 significant features (see Fig.1).

The goal of classifier is to detect and recognize attacks. There are 22 types of attacks in KDD-99 data set. All attacks can be divided into four main classes: DoS, U2R, R2L and Probe.

DoS – denial of service attack. This attack led to overloading or crashing of networks;

U2R – unauthorized access to local super user privileges;

R2L – unauthorized access from remote user;

Probe – scanning and probing for getting confidential data.

Every class consists of different attack types.

3. IDS ARCHITECTURES

Let's examine the different neural network approaches for construction of intrusion detection systems. They are based on modular neural networks. As for input data it will be used the 41 features from KDD-99 dataset, which contain the TCP-connection information. The main goal of IDS is detection and recognition of attack type. Therefore it will be used as for output data the 5-dimensional vector, where 5 is number of attack classes plus normal connection. The significant question concerning design of IDS is the following: which features are really important? We propose to use principal component analysis (PCA) neural network for important data extraction and dimensionality reduction.

The second stage construction of IDS is to detect and to recognize attacks. In this paper we propose to apply multilayer perceptron (MLP) for this purpose. Combining two different neural networks we can obtain the various IDS architectures.

Based on our previous experiments we have chosen three main and most successful models.

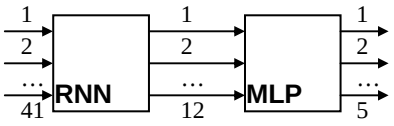

Fig. 2 – The first variant of IDS.

As shown in Fig.2 the first variant of IDS architecture consists of PCA and MLP neural networks, which are connected consequently. The PCA network, which is also called a recirculation network (RNN), transforms 41-dimentional input vector into 12-dimensional output vector. The MLP performs the processing of compressed data for recognition one type of attack or normal state.

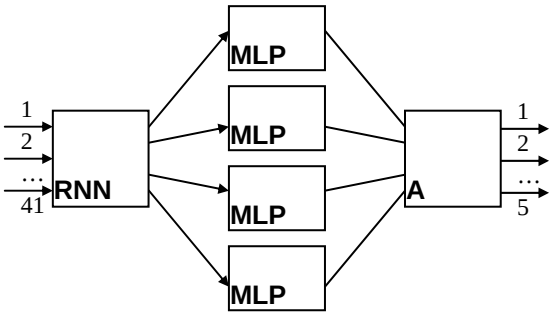

Fig. 3 – The second variant of IDS.

The second variant of IDS structure is shown in Fig.3. It consists of four MLP networks. As can be seen every MLP network is intended to recognize one type of attack: DoS, U2R, R2L or Probe. The output data from 4 multilayer perceptrons enter to Arbiter, which accept the final decision concerning type of attack. The one layer perceptron can be used as Arbiter. The training of the Arbiter is performed after leaning of RNN and MLP neural networks. Such an approach permits to fulfill the hierarchical classification attacks. In this case Arbiter can define one of 5 attack classes and corresponding MLP – type of attack.

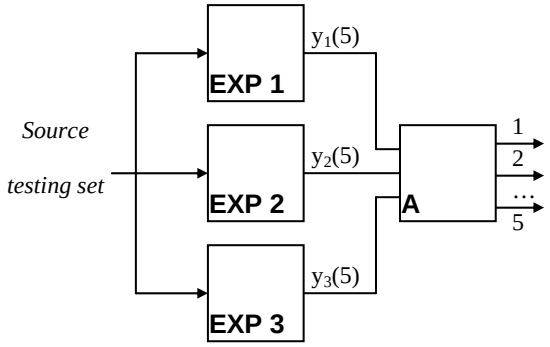

Fig. 4 – The third variant of IDS (testing mode).

Complex computational problems can be solved by dividing them into a quantity of small and simple tasks. Then the findings of each task are aggregated in general conclusion. Calculating simplicity is reached by distribution of training task among several experts. The combination of such experts (EXP) is known as Committee Machine. This integrated knowledge *per se* has priority over the opinion of each expert taking separately.

The next variant of IDS structure based on this idea is shown in Fig. 4. Expert is represented by single classification system. We use model 1 as expert. Training data set for each expert are not the same. They are organized during the training process as a result of classification performed by previous experts. The rule that was chosen for this purpose is Boosting by filtering algorithm [16]. After training the neural networks have ability to intrusion detection. In testing mode every expert is intended for processing of original 41-demensional vector. Arbiter accepts the final decision.

4. RNN NEURAL NETWORKS

In this section we present two neural networks based principal component analyses techniques, namely linear and nonlinear RNN networks.

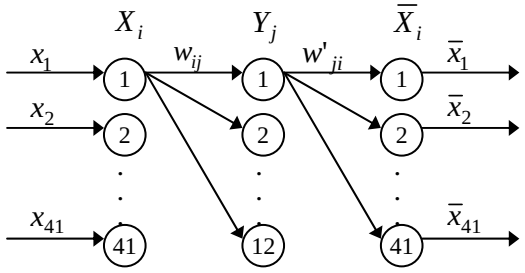

Fig. 5 – RNN architecture.

Let's consider an autoencoder, which is also called a recirculation neural network is shown in Fig.5. It is represented by multilayer perceptron, which performs the linear or nonlinear compression of the dataset through a bottleneck in the hidden layer. As can be seen the nodes are partitioned in three layers. The bottleneck layer performs the compression of the input dataset. The j-th hidden unit output in total case is given by

$$y_j = F(S_j), \tag{1}$$

$$S_j = \sum_{i=1}^{41} w_{ij} \cdot x_i, \tag{2}$$

where F is activation function; S_j is weighted sum of the j-th neuron; w_{ij} is the weight from the i-th unit to the hidden j-th unit; x_i is input to the i-th unit.

The i-th output unit is given by

$$x_i = F(S_i), \quad (3)$$

$$S_i = \sum_{i=1}^{12} w'_{ji} \cdot y_j . \quad (4)$$

In this paper we use two algorithms for RNN training. The first one is the linear Oja rule and the second one is the backpropagation algorithm for nonlinear RNN.

The weights of the linear RNN are updated iteratively in accordance with the Oja rule [17]:

$$w'_{ji}(t+1) = w'_{ji}(t) + \alpha \cdot y_j \cdot (x_i - \overline{x_i}), \quad (5)$$

$$w_{ij} = w'_{ji}.$$

As it is known such a RNN performs a linear dimensionality reduction. In this procedure the input space is rotated in such a way that the output values are so uncorrelated as possible and the energy or variances of the data is mainly concentrated in a few first principal components.

The preprocessing of input data is performed before entering it to RNN:

$$x_i^k = \frac{x_i^k - \mu(x_i)}{\sigma(x_i^k)}, \quad (6)$$

where

$$\mu(x_i) = \frac{1}{L}\sum_{k=1}^{L} x_i^k , \quad (7)$$

$$\sigma(x_i^k) = \frac{1}{L}\sum_{k=1}^{L} (x_i^k - \mu(x_i))^2 . \quad (8)$$

Here L is the number of training samples.

As it has been mentioned before the backpropagation approach for training nonlinear RNN is used. The weights are updated iteratively in accordance with the following rule:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \cdot \gamma_j \cdot F'(S_j) \cdot x_i , \quad (9)$$

$$w'_{ji}(t+1) = w'_{ji}(t) - \alpha \cdot y_j \cdot F'(S_i)(\overline{x_i} - x_i) \quad (10)$$

where γ_j is error of j-th neuron:

$$\gamma_j = \sum_{i=1}^{n} (\overline{x_i} - x_i) \cdot F'(S_i) \cdot w'_{ji} . \quad (11)$$

The weight data in the hidden layer must be reorthonormalized by using the Gram-Schmidt procedure, as follows:

1) The first vector of the orthonormal frame is chosen as:

$$w'_1 = [\frac{w_{11}}{|w_1|}, \frac{w_{21}}{|w_1|}, ..., \frac{w_{n1}}{|w_1|}], \quad (12)$$

where

$$|w_1| = \sqrt{w_{11}^2 + w_{21}^2 + ... + w_{n1}^2} . \quad (13)$$

2) The subsequent weight vector is defined by the following recurrent formulas:

$$w_i = w_i - \sum_{j=1}^{i-1} (w_i^T \cdot w'_j) \cdot w'_j , \quad (14)$$

$$|w_i| = \sqrt{w_{1i}^2 + w_{2i}^2 + ... + w_{ni}^2} , \quad (15)$$

$$w'_i = [\frac{w_{1i}}{|w_i|}, \frac{w_{2i}}{|w_i|}, ..., \frac{w_{ni}}{|w_i|}], \quad (16)$$

where i=2..12.

Let's consider the mapping of input space data for normal state and Neptune type of attack on the plane two principal components. As can be seen from the Fig. 6 the data, which belong one type of attack can be located in different areas. As a result is obtained not well the visualization of such a data using only linear RNN because of complex relationships between features. One way to decide this problem is to use the nonlinear RNN network.

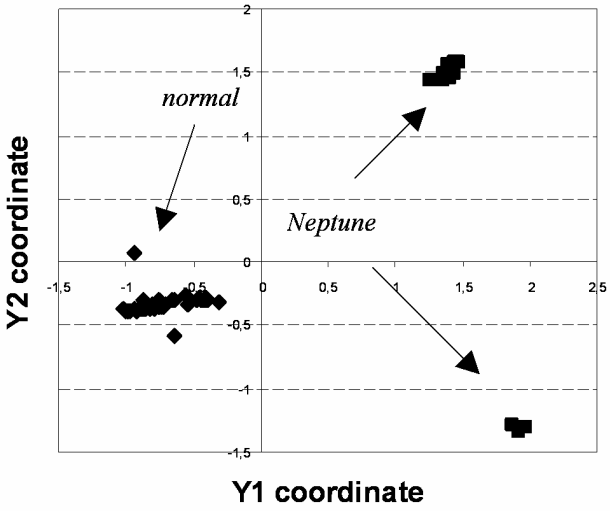

Fig. 6 – Data processed with linear RNN.

As can be seen from Fig.7 the nonlinear RNN performs the better visualization of dataset in comparison with linear RNN.

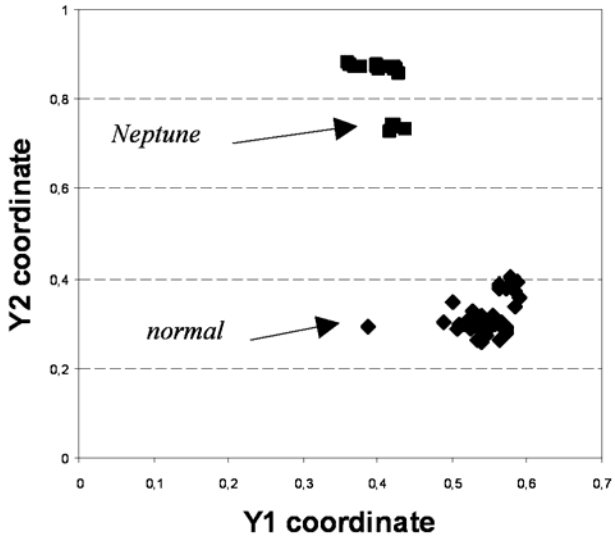

Fig. 7 – Data processed with nonlinear RNN.

5. ENSEMBLING AND MLP NEURAL NETWORKS

Let's consider the ensembling neural network. This network is trained using the boosting by filtering algorithm [16] as it is shown in Fig.8. It consists of the following steps:

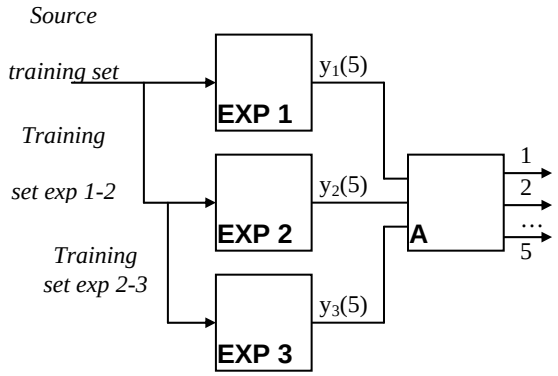

Fig. 8 – The third variant of IDS (training mode).

1) Train a first expert network using some training set;

2) A training set for a second expert is obtained in the following manner:

(a) toss a fair coin to select a 50% NEW training set and add this data to the training set for the second expert network;

(b) train the second expert;

3) A third expert is obtained in the following manner:

(a) pass NEW data through the first two expert networks. If the two experts disagree, add this data to the training set for the third expert:

(b) train the third expert network.

4) vote to committee output.

In our case the Arbiter performs vote functions. Arbiter is represented by multilayer perceptron.

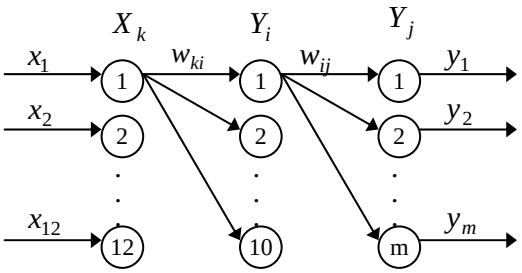

Fig. 9 – MLP architecture.

As it is mentioned before the MLP is intended for attack classification on the basis of components, which are obtained using RNN (Fig. 9). The number of output units depends on number of attack classes. The backpropagation algorithm is used for training MLP. After training of neural networks they are combined in an intrusion detection system.

6. EXPERIMENTAL RESULTS

To assess the effectiveness of proposed intrusion detection approaches, the series of experiments were performed. The KDD cup network data set was used for training and testing different neural network models, because it is one of the few in the domain of intrusion detection and it attracts the researchers' attention due to its well-defined and readily accessible nature.

The boosting by filtering algorithm, which is used in the case with model 3, needs large number of records to produce acceptable results. So we used 10% selection from KDD dataset (almost 500000 records) for testing and generation of training subset. We have used 6186 training samples for learning of neural networks. All records from 10% selection are used for testing. The same data sets were applied for model 1 and model 2. Thus we can compare the discussed models with each other. Proposed intrusion detection approaches are implemented to detect 5 classes of attacks from this dataset including DoS, U2R, R2L, Probe and Normal.

To evaluate our system we were interested in three major indicators of performance: the detection and recognition rates for each attack class and false positive rate. The detection rate (true attack alarms) is defined as the number of intrusion instances detected by the system divided by the total number of intrusion instances present in the test set. The recognition rate is defined in a similar manner. The false positive rate (false attack alarms) represents the total number of normal instances that were classified as intrusions divided by the total number of normal instances.

Let's examine the recognition of attacks with the model 1 (see Section 3). This model is quite simple. Table 1 shows statistics of recognition depending on attack class.

Table 1. Attack classification with Model 1

class	count	detected	recognized
DoS	391458	391441 (99.99%)	370741 (94.71%)
U2R	52	48 (92.31%)	42 (80.77%)
R2L	1126	1113 (98.85%)	658 (58.44%)
Probe	4107	4094 (99.68%)	4081 (99.37%)
normal state			
normal	97277	---	50831 (52.25%)

From the above results, the best detection and recognition rates were achieved for DoS and Probe connections. U2R and R2L attack instances were detected a bit worse (80.77% and 58.44% respectively). Besides, the bottom row shows that some normal instances were (incorrectly) classified as intrusions.

The number of false positives produced by previous classification model is considerable. This can be corrected by application of other models proposed in Section 3. As for model 2 (see Table 2), it performed well enough in terms of false positives due to four single multilayer perceptrons for each attack class.

Table 2. Attack classification with Model 2

class	count	detected	recognized
DoS	391458	391063 (99.90%)	370544 (94.66%)
U2R	52	49 (94.23%)	37 (71.15%)
R2L	1126	1088 (96.63%)	1075 (95.47%)
Probe	4107	3749 (91.28%)	3735 (90.94%)
normal state			
normal	97277	---	83879 (86.22%)

Model 3 (see Table 3) uses opinion of three experts. As it was mentioned above each expert is represented by a single classification system (in this experiments we use model 1 as expert). But every subsequent expert exerts influence on the outputs of others performing aggregated opinion of several neural networks.

Table 3. Attack classification with Model 3

class	count	detected	recognized
DoS	391458	391443 (99.99%)	370663 (94.69%)
U2R	52	50 (96.15%)	42 (80.76%)
R2L	1126	1102 (97.87%)	1086 (96.45%)
Probe	4107	3954 (96.27%)	3939 (95.91%)
normal state			
normal	97277	---	84728 (87.09%)

This two algorithms (model 2 and model 3) perform to each other relatively close. It was difficult to make correct comparison. But on closer examination we decided to give preference to model 3.

The total results of the detection rates and false positive rates related with each model are considered in Table 4.

Table 4. Total results for each Model

model	True attack alarms	False attack alarms	Recognized correctly	Total recog-nized %
Model 1	396696 (99.98%)	46446 (47.75%)	375522 (94,65%)	86.30%
Model 2	395949 (99.80%)	13398 (13.77%)	375391 (94.61%)	92.97%
Model 3	396549 (99.95%)	12549 (12.90%)	375730 (94.70%)	93.21%

In general, model 3 is shown to achieve the lowest false positive rates and the highest accuracy (93,21%). In fact, it is more accurate than model 1 (86.3%) and model 2 (92.97%). So model 2 and model 3 can effectively be used for classification of huge input data set with complicated structure.

Let's consider the performance of intrusion detection system with the nonlinear RNN by the example of model 1. Nonlinear RNNs were used with logical sigmoid function of activation. We applied model 1 to single services (HTTP, FTP_DATA, SMTP etc. See Table 5 and Table 6).

Table 5. Attack identification and recognition for service HTTP (with Model 1)

model	service HTTP			
	True attack alarms	False attack alarms	Recognized correctly	Total recognized %
linear RNN	2407 (100%)	470 (0.76%)	2406 (99.96%)	99.27%
nonlinear RNN	2407 (100%)	65 (0.11%)	2405 (99.92%)	99.91%

Table 6. Attack identification and recognition for service FTP_DATA (with Model 1)

model	service FTP_DATA			
	True attack alarms	False attack alarms	Recognized correctly	Total recognized %
linear RNN	893 (96.74%)	76 (2.00%)	881 (95.45%)	97.50%
nonlinear RNN	866 (93.82%)	44 (1.16%)	400 (43.34%)	87.99%

Apparently from results, the unequivocal answer to a question – what choice is better – is not present. It is desirable to use nonlinear RNN for HTTP service, in contrast to FTP_DATA, where linear RNN is preferable. Thus it is possible to draw a conclusion, in some cases it is advisable to use linear RNN and in other cases it's better to use the nonlinear RNN.

7. CONCLUSION

In this paper the neural network architectures for intrusion detection have been addressed. The proposed approach is based on integration of the recirculation network and multilayer perceptron. The KDD-99 dataset was used to perform experiments. By combining two different neural networks (RNN and MLP) it is possible to produce efficient performance in terms of detection and recognition attacks on computer networks. The main advantages of using neural network techniques are ability to recognize novel attack instances and quickness of work, which is especially important in real time mode.

8. REFERENCES

[1] S.Kumar and E.H.Spafford, "*A Software architecture to support misuse intrusion detection*", Proceedings of the 18th National Information Security Conference, pp.194-204, 1995.

[2] K.Ilgun, R.A.Kemmerer, P.A.Porras. "*State transition analysis: A rule-based intrusion detection approach*", IEEE Transaction on Software Engineering, vol.21, no.3, pp.181-199, 1995.

[3] *SNORT*, http://www.snort.org.

[4] T.Lunt, A.Tamaru, F.Gilham, et al, "*A Real-time Intrusion Detection Expert System (IDES) – final technical report*", Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, Feb. 1992.

[5] P.A.Porras and P.G.Neumann, "*EMERALD: Event monitoring enabling responses to anomalous live disturbances*", Proceedings of National Information Systems Security Conference, Baltimore MD, October 1997.

[6] D.E.Denning, "*An intrusion-detection model*", IEEE Transaction on Software Engineering. vol.13, no.2, pp.222-232, 1987.

[7] W.Lee, S.Stolfo, K.Mok, "*A data mining framework for adaptive intrusion detection*", Proceedings of the 1999 IEEE Symposium on Security and Privacy, Los Alamos, CA, pp.120-132, 1999.

[8] W.Lee, S.Stolfo, "*A Framework for constructing features and models for intrusion detection systems*", ACM Transactions on Information and System Security, vol3, no.4, pp.227-261, 2000.

[9] Y.Liu, K.Chen, X.Liao, et al, "*A genetic clustering method for intrusion detection*", Pattern Recognition, vol.37, no.5, pp.927-924, 2004.

[10] E.Eskin, A.Rnold, M.Prerau, L.Portnoy, S.Stolfo, "*A Geometric framework for unsupervised anomaly detection*", Applications of Data Mining in Computer Security. Kluwer Academics, 2002.

[11] M.Shyu, S.Chen, K. Sarinnapakorn, L.Chang, "*A Novel Anomaly Detection Scheme Based on Principal Component Classifier*", Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03), pp. 172-179, 2003.

[12] H.Kayacik, A.Zincir-Heywood and M.Heywood, "*On the capability of an SOM based intrusion detection system*", in Proc. IEEE Int. Joint Conf. Neural Networks

(IJCNN'03), pp. 1808-1813, 2003.

[13] Zheng Zhang, Jun Li, C.N. Manikopoulos, Jay Jorgenson, Jose Ucles, "*HIDE : a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification*", Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, pp.85-90, 2001.

[14] V.Golovko, L.Vaitsekhovich, "*Neural Network Techniques for Intrusion Detection*", Proceedings of International Conference on Neural Networks and Artificial Intelligence (ICNNAI-2006), pp.65-69, 2006.

[15] *1999 KDD Cup Competition*. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[16] H.Drucker, R.Schapire and P.Simard, "*Improving performance in neural networks using a boosting algorithm*", In S.J.Hanson, J.D.Cowan and C.L.Giles eds., Advanced in Neural Information Processing Systems 5, Denver, CO, Morgan Kaufmann, San Mateo, CA, pp.42-49, 1993.

[17] E. Oja, "*Principal components, minor components and linear networks. Neural Networks*", vol.5, pp.927-935, 1992.

Prof. Vladimir Golovko *was born in Belarus in 1960. He received M.E. degree in Computer Engineering in 1984 from the Moscow Bauman State Technical University. In 1990 he received PhD degree from the Belarus State University and in 2003 he received doctor science degree in Computer Science from the United Institute of Informatics problems national Academy of Sciences (Belarus). At present he work as head of Intelligence Information Technologies Department and Laboratory of Artificial Neural Networks of the Brest State Technical University. His research interests include Artificial Intelligence, neural networks, autonomous learning robot, signal processing, chaotic processes, intrusion and epilepsy detection. He has published more than 150 scientific papers, including 3 books and 2 chapters of books.*

Leanid Vaitsekhovich*, was born in Belarus in 1983. He received M.E. degree with honors in Computer Science in 2006 from the Brest State Technical University, Belarus. At present he is master student of Brest State Technical University, "System Analysis, Control and Processing of Information", department of Intelligent Information Technologies. His research interests include computer network security, data-mining, artificial intelligence and neural networks.*