EMPLOYMENT OF NEURAL NETWORK BASED CLASSIFIER FOR INTRUSION DETECTION

Leanid VAITSEKHOVICH[*] , Vladimir GOLOVKO[*]

[*]Brest State Technical University, Moskovskaja str. 267, 224017 Brest, Belarus

vspika@rambler.ru, gva@bstu.by

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 a 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. They are based on the 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 with the help of 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.

1. INTRODUCTION

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

As a result the number of attacks and criminals concerning computer networks are increasing. Therefore the security of computer networks is becoming 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 its task in real time. There exist two main intrusion detection methods: misuse detection and anomaly detection. Misuse detection is based on the already known signatures of intrusions or vulnerabilities. The main disadvantage of this approach is that it cannot detect novel or unknown attacks that were not previously defined. There are examples of misuse detection models: IDIOT (Kumar and Spafford, 1995), STAT (Ilgun et al., 1995) and Snort (http://www.snort.org). 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 (Lunt et al., 1992) and EMERALD (Porras and Neumann, 1997).

There exist different defense approaches to protect the computer networks. The principal component classifier is examined in Denning, 1987; Lee et al, 1999. The data mining techniques were presented in Lee andStolfo, 1999; Liu et al., 2004. 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) (Eskin et al., 2002; Shyu et al., 2003). Different neural networks can be used for intrusion detection (Kayacik et al., 2003; Zheng et al., 2001): 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 (Shyu et al, 2003). Therefore processing a large amount of audit data in real time is very important for practical implementation IDS.

In our previous paper (Golovko and Vaitsekhovich, 2006) 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 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.

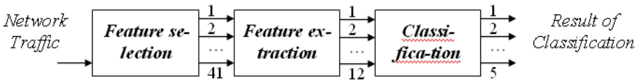

Fig. 1. The detection process

The first stage involves measurement of network traffic for feature selection. The special software monitor selects characteristics of the network traffic for features obtaining. In this paper we use the KDD-99 data set (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html). 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 flow 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 a 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, the second is a protocol type, and the third is a service name, and so on. Therefore during 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 there exist complex relationships, which are difficult to discover. Some data may be redundant and not useful for IDS. A large amount of features can increase computation time. Therefore feature extraction is a very important stage. In this paper we use linear and nonlinear PCA neural networks (RNN) for important data extraction. As a result we have extracted 12 significant features (see Fig. 1). This number has been taken through several trials where quantity of principal components incremented by one. Then the model with sufficient performance test and the smallest number of principal components was chosen.

The goal of the classifier is to detect and recognize attacks. There are 22 types of attacks in KDD-99 data set. All the attacks can be divided into four main classes: DoS, U2R, R2L and Probe.DoS – denial of service attack. This attack leads to overloading or crashing of networks; U2R – unauthorized access to local super user privileges; R2L – unauthorized access from a remote user; Probe – scanning and probing for getting confidential data.

Each 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 they will be used the 41 features from KDD-99 dataset, which contain the TCP-connection information. The main goal of IDS is the detection and recognition type of attack. Therefore the 5-dimensional vector will be used for output data, where 5 is the 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 a multilayer perceptron (MLP) is proposed to be applied for this purpose. Combining two different neural networks we can obtain the various IDS architectures.

We have chosen three main and most successful models based on our previous experiments.

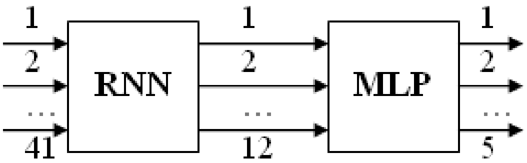

Fig. 2. The first variant of IDS

As it is 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 of one type of attack or normal state.

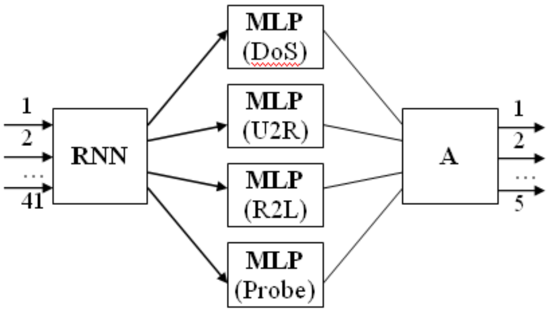

Fig. 3. The second variant of IDS

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

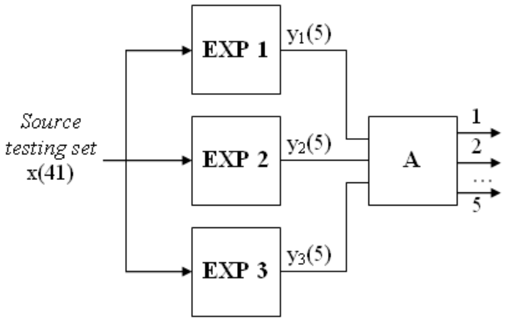

Fig. 4. The third variant of IDS (testing mode)

Complex computational problems can be solved by dividing them into a quantity of small and simple tasks. Then the results 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 a single classification system. We use model 1 as an expert. Training data set for each expert are not the same. They are organizing during the training process as a result of classification performed by the previous experts. The rule that was chosen for this purpose is Boosting by filtering algorithm (Drucker et al., 1993) which is discussed in Section 5. 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.

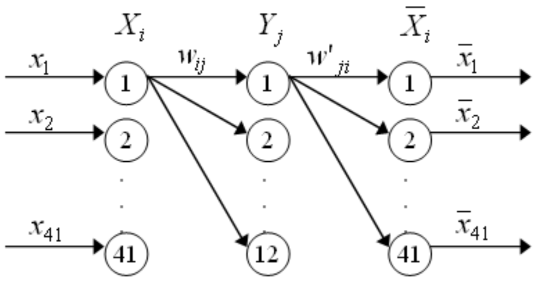

Fig. 5. RNN architecture

Let's consider an autoencoder, which is also called a recirculation neural network 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 we can see 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 – i-th unit input.

The i-th output unit is given by

$$x_i = F(S_i), \tag{3}$$

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

In this paper we use two algorithms for RNN training. One is the linear Oja rule and the other is the backpropagation algorithm for nonlinear RNN.

The weights of the linear RNN are updated iteratively in accordance with the Oja rule (Oja, 1992):

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

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

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

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

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \cdot \gamma_j \cdot F'(S_j) \cdot x_i , \tag{6}$$

$$w'_{ji}(t+1) = w'_{ji}(t) - \alpha \cdot y_j \cdot F'(S_i)(\overline{x_i} - x_i) , \tag{7}$$

where γ_j is error of j-th neuron:

$$\gamma_j = \sum_{i=1}^{n} (\overline{x_i} - x_i) \cdot F'(S_i) \cdot w'_{ji} . \tag{8}$$

The weights 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:

$$w'_1 = [\frac{w_{11}}{|w_1|}, \frac{w_{21}}{|w_1|}, \dots, \frac{w_{n1}}{|w_1|}] , \tag{9}$$

where

$$|w_1| = \sqrt{w_{11}^2 + w_{21}^2 + ... + w_{n1}^2} \tag{10}$$

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

$$w_i = w_i - \sum_{j=1}^{i-1}(w_i^T \cdot w'_j)\cdot w'_j\,, \tag{11}$$

$$|w_i| = \sqrt{w_{1i}^2 + w_{2i}^2 + ... + w_{ni}^2}\;, \tag{12}$$

$$w'_i = [\frac{w_{1i}}{|w_i|}, \frac{w_{2i}}{|w_i|}, ..., \frac{w_{ni}}{|w_i|}]\,, \tag{13}$$

where i=2..12.

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

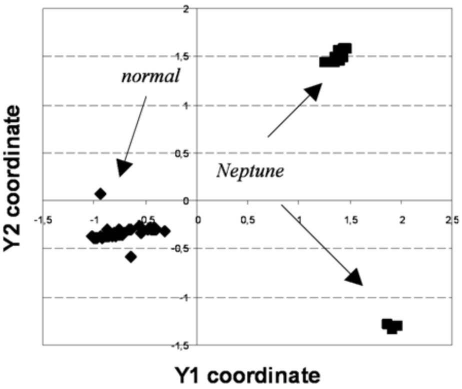

Fig. 6. Data processed with linear RNN

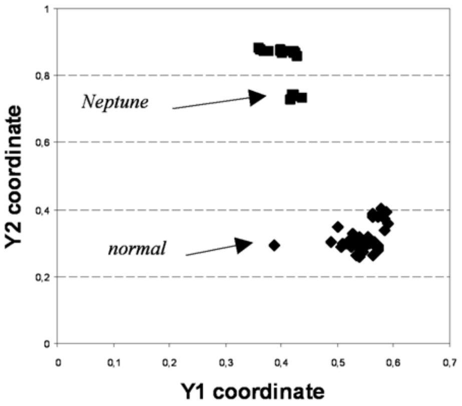

Fig. 7. Data processed with nonlinear RNN

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

5. ENSEMBLING NEURAL NETWORKS

Let's consider the ensembling neural network. This network is trained using the boosting by filtering algorithm (Drucker et al., 1993) as it is shown in Fig. 8. It consists of the following steps:

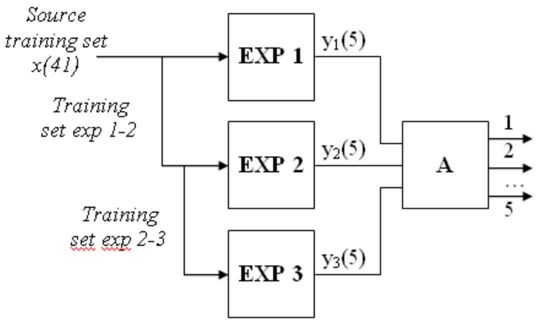

Fig. 8. The third variant of IDS (training mode)

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 way:
   (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 the multilayer perceptron.

After being trained, neural networks 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 a large number of records to produce acceptable results. So we have used 10% selection from KDD dataset (almost 500 000 records) for testing and ge-neration of training subset. We have used 6186 samples for training neural networks. All records from 10% selection are used for testing (see Tab. 1). The same data sets were applied for model 1 and model 2. Thus we can compare the discussed models. Proposed intrusion detection approaches are implemented to detect 5 classes of attacks from this dataset including DoS, U2R, R2L, Probe and Normal.

Tab. 1. Training and testing samples

	DoS	U2R	R2L	Probe	Normal	Total count
training samples	3571	37	278	800	1500	6186
testing samples	391458	52	1126	4107	97277	494020

To evaluate our system we have been 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. Tab. 2 shows statistics of recognition depending on attack class.

Table 2. 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%)

The above results show that 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 Tab. 3), it performed quite well in terms of false positives due to four single multilayer perceptrons for each attack class.

Model 3 (see Tab. 4) uses the 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 an expert). But every subsequent expert exerts the influence on the outputs of others performing aggregated opinion of several neural networks.

Table 3. 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%)

Table 4. 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 Tab. 5.

Table 5. 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 be effectively used for the classification of huge input data set with a complicated structure.

7. CONCLUSION

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

REFERENCES

1. **Denning D. E.** (1987), An intrusion-detection model, *IEEE Transaction on Software Engineering*, Vol. 13, No. 2, 222-232.
2. **Drucker H., Schapire R., Simard P.** (1993), Improving performance in neural networks using a boosting algorithm, *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, 42-49.
3. **Eskin E., Rnold A., Prerau M., Portnoy L., Stolfo S.** (2002), *A Geometric framework for unsupervised anomaly detection*, Applications of Data Mining in Computer Security, Kluwer Academics.
4. **Golovko V., Vaitsekhovich L.** (2006), Neural Network Techniques for Intrusion Detection, *Proceedings of International Conference on Neural Networks and Artificial Intelligence (ICNNAI-2006)*, 65-69.
5. **Ilgun K., Kemmerer R. A., Porras P. A.** (1995), State transition analysis: A rule-based intrusion detection approach, *IEEE Transaction on Software Engineering*, Vol. 21, No. 3, 181-199.
6. **Kayacik H., Zincir-Heywood A., Heywood M.** (2003), On the capability of an SOM based intrusion detection system, *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN'03)*, 1808-1813.
7. **Kumar S., Spafford E. H.** (1995), A Software architecture to support misuse intrusion detection, *Proceedings of the 18th National Information Security Conference*, 1995, 194-204.
8. **Lee W., Stolfo S.** (2000), A Framework for constructing features and models for intrusion detection systems, *ACM Transactions on Information and System Security*, Vol. 3, No. 4, 227-261.
9. **Lee W., Stolfo S., Mok K.** (1999), A data mining framework for adaptive intrusion detection, *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, Los Alamos, CA, 120-132.
10. **Liu Y., Chen K., Liao X.** (2004), A genetic clustering method for intrusion detection", *Pattern Recognition*, Vol. 37, No. 5, 927-924.
11. **Lunt T., Tamaru A., Gilham F.** (1992), *A Real-time Intrusion Detection Expert System (IDES) – final technical report*, Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, Feb.
12. **Oja E.** (1992), Principal components, minor components and linear networks. Neural Networks", Vol. 5, 927-935.
13. **Porras P. A., Neumann P. G.** (1997), EMERALD: Event monitoring enabling responses to anomalous live disturbances, *Proceedings of National Information Systems Security Conference*, Baltimore MD.
14. **Shyu M., Chen S., Sarinnapakorn K., Chang L.** (2003), 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)*, 172-179
15. **Zhang Z., Li J., Manikopoulos C. N., Jorgenson J., Ucles J.** (2001), 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, 85-90.
16. *1999 KDD Cup Competition*. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
17. *SNORT*. Available: http://www.snort.org.