computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

IMPLEMENTATION OF NEURAL NETWORKS AND BOOSTING ALGORITHMS FOR EFFECTIVE INTRUSION DETECTION

Vladimir Golovko [1], Leanid Vaitsekhovich [2]

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

Abstract: *In this article the classification task in the domain of intrusion detection is considered. Often a chosen algorithm is not good enough for practical use. So the question arises how is it possible to improve the performance? In this case we can employ so-called Committee Machines that increase accuracy and reliability of the base classification model. These advantages are the result of dividing complex computational problems among several experts. The knowledge of each expert influences on the general conclusion of Committee Machine.*

Keywords: *neural networks, intrusion detection, computer security, boosting algorithms, AdaBoost.*

1. INTRODUCTION

Efficient information exchange has become the primary attribute of successful activity in any field. Recently computer technologies (computer networks, electronic commerce, corporate networks and web sites etc) performed the break in this sector. However together with growing necessity of increasing communication reliability and carrying capacity a crucial question of information resources protection has arisen [1].

There exist different defense approaches to protect the computer systems. All approaches can be divided into two main groups: organizational and technical. Technical approaches consist of network and hostbased approaches. In this article we will discuss network security tools namely intrusion detection systems.

The aim of *Intrusion Detection Systems (IDS)* is detecting inappropriate, incorrect or anomalous activity in computer systems or computer networks.

There are a lot of different means to protect computer networks: correct policy of security, gateway filters, anti-virus software etc. But as a rule IDS is assigned the role of a basic element of protection. IDS are used for early notification about network problems because generally they are allocated at a network level where suspicious actions can be found out earlier, then at higher levels. Besides IDS is able to gather necessary evidences of malicious activity as well as to reveal latent tendencies. This becomes possible due to analysis of plenty of the data.

The major problems of existing models are 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. Therefore processing a large amount of audit data in real time is very important for practical implementation of IDS. It is difficult to eliminate stated disadvantages using only classical computer security methods. Therefore IDS have been close studied recently.

This article is an extension of the previous work [2, 3, 4] associated with the development of intrusion detection system with the neural network classifier. Classification is the main problem in the intrusion detection domain. In the article some static structures for boosting are considered that theoretically make it possible to increase accuracy and reliability of recognition process.

Another reason for investigation of ensembles of classifiers in computer security area is spreading of multiprocessor computer systems that can be used for performance optimization.

The paper is organized as follows. The main conception of Committee Machines and some aspects of mixing expert decisions are given in Section 2. In Section 3 and 4 Boosting by Filtering and AdaBoost algorithms are described [5, 6]. Section 5 describes architectures of the ensemble neural networks for intrusion detection. Section 6 presents experimental results. Finally, concluding

remarks are made in the last section.

2. WHAT IS COMMITTEE MACHINE

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 a general conclusion. Calculating simplicity is reached by distribution of a training task among several *experts (EXP)*. The combination of such experts is known as *Committee Machine*. This integrated knowledge per se has priority over the opinion of each expert taking separately.

There are two main classes of Committee Machines: static and dynamic. In the case of dynamic machines input data directly influence on the generalized solution.

In this article we discuss two methods of the static algorithms for classification task in the domain of intrusion detection. The first one is *Boosting by Filtering algorithm.* And the second – *algorithm AdaBoost*, that is more perfect boosting algorithm. It has received the name due to ability to adapt to mistakes of separate experts. Both algorithms have been known for a long time.

On the whole, boosting is a general method for improving the accuracy of any given algorithm. Boosting algorithms use several "weak" algorithms (experts) that are trained on different sets of examples. The accuracy of the expert is required to be just slightly better then 1/2, which is the accuracy of a completely random guess.

The major difficulty is that it is important to detect optimal combination of parameters of certain classifier on the one hand and the whole ensemble entirely – on the other hand.

In the next few sections we will discuss mentioned above boosting algorithms in details.

3. BOOSTING BY FILTERING ALGORITHM

The *Boosting by Filtering algorithm* takes as input a training set of m examples $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, where x_i – is a single instance from the input space X (typically, a vector of attribute values), and y_i – is a label from the space Y. This algorithm uses knowledge of three experts $t=1..3$.

The algorithm of training consists of the following steps:

The BOOSTING BY FILTERING algorithm:

1. Train a first expert using training set of m examples;

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;

(b) train the second expert;

3. A third expert is obtained in the following way:

(a) pass NEW data through the first two experts.

If the two experts disagree, add this data to the training set for the third expert:

(b) train the third expert.

4. Vote to committee output. In the case of two classes:

$$H(x) = sign\left(\sum_{t=1}^{3} h_t(x)\right) \quad (1)$$

One of the main disadvantages of Boosting by Filtering algorithm is that it needs a large number of records to produce acceptable results.

4. ADABOOST ALGORITHM

The AdaBoost algorithm was proposed by Robert Schapire in 1995. This algorithm made it possible to overcome some difficulties arising with the earlier boosting algorithms. The main advantages of AdaBoost (as compared with the previous model of boosting) are:

- the number of experts is unlimited;
- the only training set can be used for the whole of ensemble of experts.

AdaBoost algorithm takes as input a training set of m examples $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, where x_i – is a single instance from the input space X, and y_i – is a label from the space Y associated with x_i. AdaBoost has access to another specified learning algorithm ("weak" learning algorithm or expert) repeatedly in a series of iterations $t=1, 2, \ldots, T$. The main idea of the algorithm is to provide each x_i from a training set with a certain numerical value – weighting coefficient, i.e. to specify distribution D on the set of examples x_i. Weighting coefficient for sample x_i on the iteration t is denoted $D_t(i)$. On the first step, all weights are set equally, but on the next iterations their values are modified so that the weights of incorrectly classified examples are increased. Thus, AdaBoost focuses the most weight on the examples which seem to be hardest for weak learning algorithm.

The ADABOOST algorithm:

1. The algorithm takes as input a training set $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ with labels y_i, $i=1..k$. Initially all weights are set equally: $D_1 = 1/m$. Steps 2-4 are repeated for each iteration $t = 1, 2, \ldots, T$.

2. Train weak learner using distribution D_t, get weak hypothesis with the weighted training error

$$e_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i) \quad (2)$$

3. Set

$$a_t = \frac{1}{2} \cdot \ln\left(\frac{1-e_t}{e_t}\right) \quad (3)$$

3. Update weights

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-a_t} & if\ (h_t(x_i) = y_i) \\ e^{a_t} & if\ (h_t(x_i) \neq y_i) \end{cases} \quad (4)$$

where Z_t – is a normalization constant, such that $\sum_i D_{t+1}(i) = 1$

4. Output the final hypothesis:

$$H(x) = sign\left(\sum_{t=1}^{T} a_t h_t(x)\right) \quad (5)$$

for two classes case, and

$$H(x) = \arg\max_{y \in \{1,2,\ldots,k\}} \left[\sum_{t:h_t(x)=y} a_t\right] \quad (6)$$

for *k* classes.

The aim of the weak learner is to compute a classifier or hypothesis h_t: *X -> Y* appropriate for the D_t. The efficiency of a weak hypothesis is measured by its error

$$e_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i) \neq y_i} D_t(i) \quad (7)$$

We can see that this error is measured with respect to the distribution Dt that was provided to the weak learner. Many learning algorithms can be modified to handle examples that are weighted by a distribution such as the one created by the boosting algorithm. When this is possible, the booster's distribution Dt is supplied directly to the weak learning algorithm. However, some learning algorithms require an unweighted set of examples. For such a weak learning algorithm, it is possible to choose a subset of examples from initial set according to the distribution Dt (for instance, with maximal values of Dt(i)).

Once the weak hypothesis ht has been received, AdaBoost chooses parameter at. Intuitively, at measures the importance that is assigned to ht, i.e. trustworthiness of a single expert conclusion. Note that at>=0, if et<=1/2, and that at gets larger as et gets smaller.

To compute distribution Dt+1 from Dt, we use the rule shown in the Equation (4). The weights are then renormalized by dividing by the normalization constant Zt. Effectively, "easy" examples that are correctly classified by many of the previous weak hypotheses get lower weight, and "hard" examples which tend often to be misclassified get higher weight. So it is possible to concentrate on "hard" examples.

5. USING BOOSTING ALGORITHMS WITH NEURAL NETWORKS

In the Section 2 and 3 boosting algorithms are discussed. Concerning the experts applied in this algorithms we suppose that they have error only slightly better than 1/2. For binary classification problems, this means that the weak hypotheses need be only slightly better than random. No other conditions are imposed.

Let's consider how to employ artificial neural networks for intrusion detection. Every pattern x_i on the input of the model is described by 41 features and labeled with y_i. Using y_i we can consider a pattern either as an attack or a non-attack.

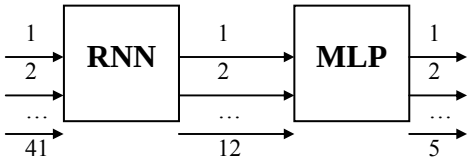

Fig. 1 – Basic classification model

This Intrusion Detection System architecture (Fig. 1) consists of PCA (Principal Components Analysis) and MLP (Multilayer Perceptron) 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 [7], which presents 12 principal components. The MLP performs the processing of compressed data for recognition of one class of attack or normal state.

There is a problem in Principal Components Analysis. We do not know the number of principal components (*p*).

It is possible to use the criterion of completeness for determination of *p*:

$$I = \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_p}{\lambda_1 + \lambda_2 + \ldots + \lambda_{41}} \quad (8)$$

where λ_i – is eigenvalue.

Our experiments (Table 1) show that the optimal number of principal components lies near 12. Other experiments confirm our results. In this work [8] the authors use 13 principal components similarly.

Table 1. Recognition rates for some set of samples depending on number of principal components

Number of principal components	Recognition rates
2	39,24%
4	47,15%
5	71,84%
7	78,16%
10	95,25%
15	96,84%
20	96,52%
41	96,84%

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.

Such intrusion detection model produces weak hypothesis h_t that influences on the generalized solution of an ensemble of experts.

The model that corresponds to Boosting by Filtering algorithm is shown on Fig. 2. The Arbiter performs vote functions and accepts the final joint resolution H of three experts. Arbiter is represented by the two-layer perceptron.

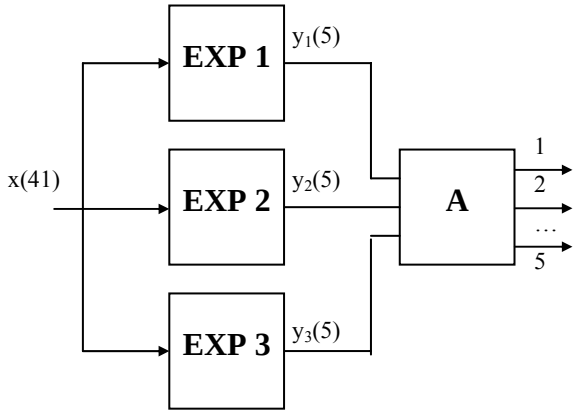

Fig. 2 – Model based on Boosting by Filtering algorithm

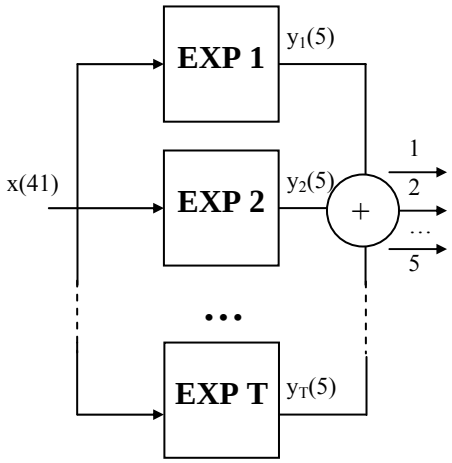

Fig. 3 – Model based on AdaBoost algorithm

In the case of AdaBoost algorithm (Fig. 3) Summator performs the functions of the Arbiter. This analog of the Arbiter generates the result of the voting by summarizing private decisions. In this procedure we take into consideration the weight coefficients of each expert.

6. EXPERIMENTAL RESULTS

In this section the results of the network traffic classification are described. We have already introduced in the previous sections three models for this purpose: i) single expert consists of PCA and MLP neural networks, which are connected consequently, ii) the Boosting by Filtering algorithm and iii) the AdaBoost algorithm.

We used data from Table 2 in the experiments for detecting intrusions.

Table 2. Training and testing samples

	DoS	U2R	R2L	Probe	Norm.	total
training samples	3571	37	278	800	1500	6186
testing samples	391458	52	1126	4107	97277	494020

The training samples were used to adapt the neural networks. After the training step the intrusion detection performance of each model was examined by the tasting samples. To evaluate our models we have been interested in three major indicators: the detection and recognition rates for each attack class and false positive rate. The results of our experiments are shown in Tables 3-5.

Table 3. Test performance of RNN+MLP model

class	class	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	97277	---	50831 (52.25%)
total	494020	---	426353 (86,30%)

Table 4. Test performance of boosting by filtering model

class	class	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	97277	---	84728 (87.09%)
total	494020	---	460458 (93,21%)

Ensembles of classifies provide ample opportunities for adjustment of neural network settings (such as amount of neurons in hidden layers, number of training cycles etc). On the one hand, this makes it possible to get required results. But then it considerably complicates construction of models, because you need a lot of experiments to select an optimal configuration.

Table 5. Test performance of AdaBoost model

class	class	detected	recognized
DoS	391458	389917 (99.61%)	369088 (94.29%)
U2R	52	51 (98.08%)	44 (84.62%)
R2L	1126	1119 (99.37%)	636 (56.48%)
Probe	4107	3908 (95.15%)	3668 (89.31%)
normal	97277	---	77212 (79.37%)
total	494020	---	450648 **(91,22%)**

Comparing results in the given tables we can conclude that boosting methods make it possible to decrease the number of false positives. The normal examples that belong to class "normal" represent a sufficiently wide range of normal connections in a computer network. So there are a lot of problems with their identification. In addition, the results of training produced by Committee Machine are more stable owing to suppression of failed weak hypotheses.

In the case of AdaBoost algorithm we have tried models with different number of experts (3, 5, 7 and 9). The general tendency is like that – one "leading" expert (or a group of such experts) appears to be very good for classification of examples from a training set. This expert (-s) has high value of parameter a_t and (in compliance with the Equations (5, 6)) significantly influences on the final decision H of the ensemble of classifiers. If this expert fails to recognize some attack class then it negatively affects on joint resolution.

To overcome this disadvantage, each neuron of the output layer was provided with a certain value q_{ty} from the range [0..1] (so-called confidence factor). This factor was calculated for a trained neural network. Examples from the training set were used, so that:

- for q = 1, the neuron responds only to samples of the same class;
- for q = 0, the neuron responds to all samples from the training set.

The common decision is calculated from the Equation 9 (compare with the Equation 6).

$$H(x) = \arg\max_{y\in\{1,2,\ldots,k\}} \left[\sum_{t:h_t(x)=y} a_t \cdot q_{ty} \right] \qquad (9)$$

Such technique is acceptable only in the case of an ensemble of classifies.

7. CONCLUSION

The experimental results show that application of ensembles makes it possible to improve some parameters of the model efficiency. However it is a result of increasing computational complexity. Besides boosting algorithms do not always result in improvement of single class recognition. Along with boosting, processes of averaging of result on an ensemble are observed. Therefore in each concrete case it is important to solve separately, whether it is necessary to apply boosting algorithm.

REFERENCES

[1] Web Application Security Consortium. Classification of security threats. – Information on: www.webappsec.org.

[2] V. Golovko and L. Vaitsekhovich. Neural Network Techniques for Intrusion Detection // In Proceedings of the International Conference on Neural Networks and Artificial Intelligence (ICNNAI-2006) / Brest State Technical University – Brest, 2006. – P. 65-69.

[3] V. Golovko, P. Kachurka and L. Vaitsekhovich. Neural Network Ensembles for Intrusion Detection // In Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-2007) / Research Institute of Intelligent Computer Systems, Ternopil National Economic University and University of Applied Sciences Fachhochschule Dortmund – Dortmund, Germany, 2007. – P. 578-583.

[4] V. Golovko, L. Vaitsekhovich, P. Kochurko and U. Rubanau. Dimensionality Reduction and Attack Recognition using Neural Network Approaches // Proceedings of the Joint Conference on Neural Networks (IJCNN 2007), Orlando, FL, USA – IEEE Computer Society, Orlando, 2007. – P. 2734-2739.

[5] 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. – 1993. – P. 42-49.
[6] Yoav Freund, Robert E. Schapire. A short introduction to boosting // Journal of Japanese Society for Artificial Intelligence. – 1999. – №14(5). – P. 771-780.
[7] V. Golovko. Neural Networks: training, organization and application. Book 4: Tutorial for students / Edited by A. Galushkin. – M.: IPRJR, 2001. – P. 256.
[8] V. Venkatachalam, S. Selvan. Performance comparison of intrusion detection system classifiers using various feature reduction techniques // I.J. of SIMULATION. – Vol. 9 No 1 – P. 30-39.

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 PhD student of Brest State Technical University, "Computing Machinery and Systems", department of Intelligent Information Technologies. His research interests include computer network security, data-mining, artificial intelligence and neural networks.*