Ensemble Learning for Intrusion Detection in Computer Networks

Luca Didaci, Giorgio Giacinto and Fabio Roli
Department of Electrical and Electronic Engineering - University of Cagliari, Italy
Piazza D'Armi - 09123 Cagliari, Italy
{luca.didaci,giacinto,roli}@diee.unica.it

Abstract. The security of computer networks plays a strategic role in modern computer systems. In order to enforce high protection levels against threats, a number of software tools are currently developed. Intrusion Detection Systems aim at detecting intruder who eluded the "first line" protection. In this paper, a pattern recognition approach to network intrusion detection based on ensemble learning paradigms is proposed. The potentialities of such an approach for data fusion and some open issues are outlined.

Keywords: Intrusion Detection, pattern classification, ensemble learning, data fusion

1 Introduction

Nowadays an increasing number of commercial and public services are offered trough Internet, so that security is becoming one of the key issues. The so-called "attacks" to internet service providers are carried out by exploiting unknown weaknesses or bugs always contained in system and application software [1-2]. Computer networks are usually protected against attacks by a number of access restriction policies that act as a coarse grain filter. Intrusion detection systems (IDS) are the fine grain filter placed inside the protected network, looking for known or potential threats in network traffic and/or audit data recorded by hosts.

Two approaches to intrusion detection are currently used. The first one, called *misuse* detection, is based on attack *signatures*, i.e., on a detailed description of the sequence of actions performed by the attacker. This approach allows the detection of intrusions matching perfectly the signatures. The effectiveness is strictly related to the extent to which IDSs are updated with the signatures of the latest attacks developed. This is currently a challenge since new attacks and new attack variants are constantly being developed. In particular, at the time an attack signature is made publicly available, a number of attack variants are designed to produce the same effect as the original attack, but with a slightly different signature that is not detected by signature-based IDSs. The development of signatures with a limited scope is motivated by the following reasons: i) the difficulty in capturing the "root-cause" of an attack and ii) the requirement of a very small false alarm rate. If the signatures are too general, then high attack detection rates may be associated with unacceptable false alarm rates, as a relevant number of normal traffic events may match the signature. It is easy to see that the availability of signatures coding the "root-cause" of the attacks should protect from all attacks exploiting the same vulnerability. Unfortunately this is beyond the current state of the art of IDSs [2].

The second approach is based on statistical knowledge about the *normal* activity of the computer system, i.e., a statistical profile of what constitutes the legitimate traffic in the network. In this case, intrusions correspond to *anomalous* network activity, i.e. to traffic whose statistical profile deviates significantly from the normal one [1-2]. This IDS model is capable of detecting intrusions regardless of the system type, the environment, the system vulnerabilities, and the type of intrusions, provided that the set of usage profiles stored in the system actually represents its "normal" working condition, and that any attack against the system involves its abnormal use. Unfortunately, the acquisition of profiles of "normal" activity is not an easy task. First of all, the audit records used to produce the profiles of normal activity may contain traces of intrusions not detected by the IDS. In this case intrusions are considered legitimate activities. In addition, imperfect normal profiles may cause a high rate of false alarms when the activity of legitimate users deviates from what is stored as being "normal"

The above discussion points out that the two intrusion detection approaches are usually formulated in terms of "pattern matching" paradigms. "Anomaly detection" approaches classify as being intrusive all the activities not matching the normal profiles. On the other hand "misuse detection" approaches classify as being intrusive the activities matching one of the attack profiles. While pattern matching is effective when the objects to be classified exhibit quite a regular structure, this does not seem to be the most common case for current IDSs, because the development of pattern matching rules for both anomaly and misuse detection relies on the experience and intuition of human experts [3]. As a consequence, as pointed out in the above, such rules can hardly adapt to the high variability of "normal" activities or to the number of novel attacks constantly being developed.

These difficulties in current IDSs lead researchers to apply statistical pattern recognition approaches based on the learning by examples paradigm [4]. The main motivation for using pattern recognition approaches for the development of advanced IDSs is their generalization capability, which may support the recognition of intrusions that have not been seen previously and have no previously described patterns.. This formulation of intrusion detection problem combines the advantages of signature-based and anomaly-based IDS. At present research on advanced IDSs based on learning by example paradigms is at an early stage, so that a number of issues should be solved in order to be used in operational environments [3]. However the few works in the literature showed promising results, thus calling for further research [5-11].

In this paper, an approach to intrusion detection in computer networks based on the ensemble learning paradigm is proposed [12]. Each member of the ensemble is trained on a distinct feature representation of patterns, then the results of the ensemble members are combined. This approach is motivated by the observation that human experts combine attack evidence from different feature sets to code attack signatures. A review of the current state of the art on IDSs based on learning by example paradigms is given in Section 2. Section 3 presents a formulation of the intrusion detection problem as a pattern recognition task. The data fusion approach based on the ensemble learning paradigm is illustrated in section 4, and results on a public available data set are reported in Section 5. Conclusions are drawn n Section 6.

2 Related Work on Pattern Recognition Approaches to Intrusion Detection

A recent technical report on current Intrusion Detection (ID) technology, where both commercial and research products are briefly reviewed, provides a discussion on the challenges to developing effective IDSs [3]. In particular it has been pointed out that advanced research issues on IDSs should involve the use of pattern recognition and learning by example approaches for the following three main reasons:

- the capability of learning by example approaches to generalize from a representative set of examples, allows to detect new types of intrusion;

- with learning by example approaches attack "signatures" can be extracted automatically from labeled traffic data, thus allowing to overcome the subjectivity of human interpretation of intrusive behavior, the latter being implemented in many current IDSs;
- learning by example approaches are able to adapt to new threats.

These issues have been addressed ever since the early years of IDS development. In particular the application of neural networks for IDSs has been investigated by a number of researchers. Neural networks provide a solution to the problem of modelling the users' behavior in anomaly detection because they do not require any explicit user model [5-7]. Neural networks for intrusion detection have first been introduced as an alternative to statistical techniques in the IDES intrusion detection expert system to model users' behavior [5]. In particular the typical sequence of commands executed by each users is learned. Similar approaches have been presented later [6]. A different approach to anomaly detection based on neural networks is proposed in [7]. While previous works have addressed the anomaly detection problem by analyzing the audit records produced by the operating system, in [7] anomalies are detected by looking at the usage of network protocols.

A neural network model designed to perform anomaly and misuse detection has been proposed in [8]. The training set is made up of strings of events captured by the Base Security Module (BSM) that is part of many operating systems. If the training set is made up of strings related to normal behavior, neural networks act as an anomaly detector. On the other hand, if strings captured during an attack session are included in the training set, the network model can be modified to act as a misuse detection system.

Training sets made up of traffic data instead of audit records have also been used for misuse detection [9-10]. Traffic features at different levels of abstraction have been used, from packet data [9] to very high level features, such as the security level of the source and destination machines, occurrences of suspicious strings, etc. [10]. For each neural network model, different numbers of output nodes have been selected according to the attack classification used. In some cases one network for each attack class has been used.

The above discussion points out that pattern recognition techniques based on neural networks are apt to provide a solution to some open issues in IDS development, e.g. the automatic extraction of normal and attack signatures from data and the ability to detect attacks not known at training time. In addition, the extensive evaluation of pattern classification techniques carried out on a sample data set of network traffic performed during the KDD'99 conference pointed out the feasibility of a pattern recognition approach to ID [11].

A common characteristic shared by all the pattern recognition approaches reviewed is the use of a feature space made up of features related to information at different abstraction levels. Except for the neural network hierarchy proposed in [7] where features at different abstraction levels are processed by distinct networks, the classification is usually performed in the feature space made up of all the features needed to detect the considered attack classes. Recent advances in pattern recognition theory suggest that classification performances may be improved by combining an ensemble of pattern classifiers trained on different feature representations [12,13]. First a decision is made by individual classifiers on the basis of partial information, e.g. intrinsic information, knowledge of past events, etc. Then the decisions are combined by means of suitable decision combination functions. Performances of ensemble approaches are usually higher than those of individual pattern classifiers. In addition, a combined system exhibits higher robustness with respect to uncertainties in training data. This is an important feature in IDSs because of the difficulties collecting a representative set of examples. In the following section a formulation of the intrusion detection task as a pattern recognition problem is proposed. In section 4 an approach to intrusion detection based on ensemble learning paradigms is presented.

3 Problem Formulation

Intrusions in computer networks basically exploit two different kinds of weaknesses: weaknesses of the network transmission protocol (network-based intrusions) and weaknesses and bugs exhibited by system and application software (host-based intrusions). As the vast majority of host based information can be extracted from the network traffic flow, intrusion detection systems based on network traffic processing are capable of detecting not only intrusion based on the exploitation of weaknesses of network protocols but also intrusions based on weaknesses of system and application software. Accordingly, the proposed pattern recognition approach will be focused on network intrusion detection.

From the pattern recognition point of view, the network intrusion detection problem can be formulated as follows (see Figure 1): given the information about network connections between pairs of hosts, assign each connection to one out of *N* data classes representing normal traffic or different categories of intrusions (e.g., Denial of Service, access to root privileges, etc.). It is worth noting that various definitions of data classes are possible [1-2].

The term "connection" refers to a sequence of data packets related to a particular service, e.g., the transfer of a web page via the http protocol. As the aim of a network intrusion detector is to detect those connections that are related to malicious activities, each network connection can be defined as a "pattern" to be classified. This formulation is in agreement with the related work presented in Section 2.

The extraction of suitable features representing network connections is based on expert knowledge about the characteristics that distinguish attacks from normal connections. On the basis of previous work on feature extraction for intrusion detection, the following three main feature sets can be used to classify each connection [14].

- intrinsic features, i.e., general information related to the connection. They include the duration, type, protocol, flag, etc. of the connection;
- traffic feature, i.e., statistics related to past connections similar to the current one e.g., number of connections with the same destination host or connections related to the same service in a given time window or within a predefined number of past connections;
- content features, i.e., features containing information about the data content of packets (“payload”) that could be relevant to discover an intrusion, e.g., errors reported by the operating system, root access attempts, etc.

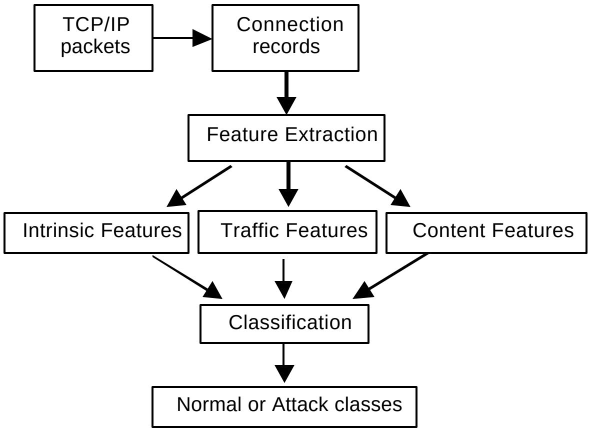

Figure 1. Intrusion Detection formulated as a Pattern Recognition Problem

In our opinion, the proposed feature categorization is general enough to take into account the variability of network traffic and the different abstraction levels required to detect attacks. In addition it is worth noting that all the features used by current IDSs fall into one of the above feature categories.

Usually a hand-coded signature involves the evaluation of a set of features belonging to different feature categories. Except for very simple attacks, the choice of the more suitable features for signature-based attack detection is not an easy task, as "traces" of an attack can usually be detected by looking at different features. The choice depends essentially on human expert knowledge and intuition about the effectiveness of the signature in terms of high attack detection accuracy and low false alarms rates [3]. On the other hand the use of a pattern recognition approach may exploit the richness of information provided by all the available features extracted from network traffic data. More complex signatures may be coded, thus allowing for more effective separation of attacks from normal traffic. As a consequence, a pattern recognition approach may provide higher detection accuracy than by hand-coded signatures. In addition the generalization capability of pattern recognition algorithms allows for the detection of novel attacks.

4 A classifier ensemble approach

In the previous section we have pointed out that three types of features can be extracted from network traffic data. Each feature category provides information that can be used to discriminate between attacks and normal traffic. In particular, when an attack is performed against a computer network, a "signature" related to that attack may be found in each feature category. As these "signatures" exhibit different discriminative power with respect to attack detection, network analysts produce an effective attack "signature" by selecting the more effective subsets of features according to their experience and intuition. On the other hand, pattern recognition tools have been designed to process the entire available feature set to extract more effective signatures than the ones hand-coded by network analysts (see Section 2). A pattern recognition approach based on the ensemble learning paradigm may exploit the above observation that attack evidence may be collected separately in different feature subspaces. While the original ensemble method is based on averaging posterior estimates of multiple classifiers, each classifier providing a different solution to the same problem, more recently other paradigms have been added to the pool [15]. A sufficient and necessary condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate (i.e., accuracy higher than 0.5) and diverse (i.e., they make different errors on new data points). In order to generate ensembles made up of accurate and diverse classifiers, a number of methods have been proposed [15]. Some of them are based on re-sampling methods (e.g., Bagging and Boosting), so that each ensemble member is trained on a different training set. Other methods are based on training a number of classifiers using the same training sets but with different subsets of features. It is easy to see that this approach is well suited when the data at end are described by distinct subsets of features, as in the Intrusion Detection problem. It is worth noting that in this case the goal is not only to improve the performances with respect to individual ensemble members, but also to improve the performances with respect to classifiers trained on the entire feature space.

The ensemble method proposed for solving the Intrusion Detection problem can be illustrated as follows. First each feature subspace is used independently to perform attack detection. Then the evidences are combined in order to produce the final decision (see Figure 2). This process reflects the human analyst perspective that usually looks at different traffic statistics in order to produce reliable attack signatures, i.e. signatures providing effective attack detection and a very low false positive rate. The approach based on ensemble learning may also attain effective attack detection as the combination of multiple evidences usually exhibits higher accuracies, i.e. lower false positives, than individual decisions. In addition, the generalization capabilities of pattern recognition algorithms allow for the detection of novel attacks that is not provided by rule-based signatures.

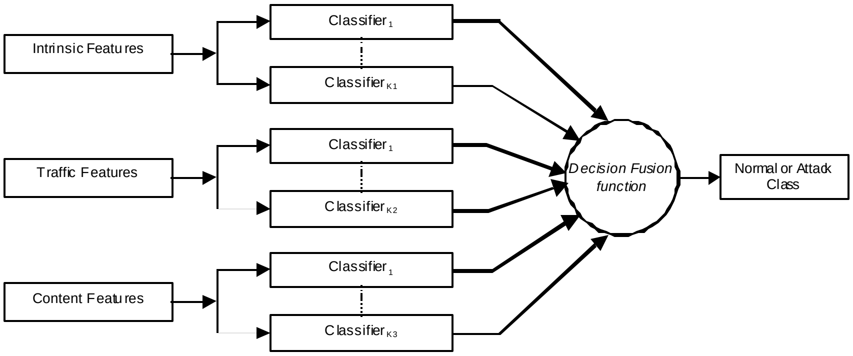

Figure 2. Ensemble of Pattern Classifiers for Intrusion Detection

In order to illustrate this approach, we used three simple fusion techniques: the majority voting rule, the average rule and the "belief" function. These fusion techniques compute the final decision from the set of decisions of an ensemble made up of K classifiers. The "majority voting rule "assigns a given input pattern to the majority class among the K outputs of the classifiers combined. The "average rule" assigns a given input pattern to the class with the maximum average posterior probability, the average being computed among the K classifiers (this rule can be applied if classifiers provide estimates of posterior probabilities, like multi-layer perceptron neural networks). The third fusion rule is based on the computation of a "belief" value for each data class given the set of outputs of the K classifiers. Belief values are based on estimates of the probabilities that a pattern assigned to a given data class actually belongs to that class or to other classes. These probabilities can be easily computed from the confusion matrix on the training set. The classification is then performed by assigning the input pattern to the data class with the maximum "belief" value. For more details about the above combination methods the reader is referred to [16].

These methods for combining the outputs of the classifier ensemble are based on the assumption that the outputs of different classifiers are independent. This assumption cannot be verified in practice, even if in many real cases this assumption may reasonably hold. In particular this assumption may reasonably hold for the problem at hand, since the three feature sets are related to independent connection characteristics. For example, for a given set of values of the intrinsic features (e.g., the number of bytes transmitted), there is no relationship with the values assumed by the content features (e.g., an attempt to log in the system as user "root"). For this reason it can be argued that the combination of classifiers trained on different feature sets could provide better performances than each single classifier. In addition, performances should also be higher than those of classifiers based on a single feature vector containing all the available features.

5 Experimental results

Experiments have been carried out on a subset of the data base created by DARPA in the framework of the 1998 Intrusion Detection Evaluation Program [17]. This subset have been pre-processed by the Columbia University and distributed as part of the UCI KDD Archive [18]. The available data base is made up of a large number of network connections related to normal and malicious traffic. Each connection is represented with a 41 dimensional feature vector according to the set of features illustrated in section 2. Connections are also labeled as belonging to one out of five classes, i.e., normal traffic, Denial of

Service (DoS) attacks, Remote to Local (R2L) attacks, User to Root (U2R) attacks, and Probing attacks. Each attack class is made up of different "attack types", i.e. attacks designed to attain the same effect by exploiting different vulnerabilities in computer networks. It is worth noting that these features are currently extracted from network traffic by a large number of commercial and open-source sniffers and IDSs. Consequently the proposed ensemble approach is well suited to be used in a real scenario.

In order to test our pattern recognition approach, we restricted our investigation to connections related to the *ftp* service. These connections can be represented by a feature set containing 30 out of the 41 available features of the data set. It was possible to discard 11 features out of 41 because they exhibit a constant value over all *ftp* connections (these features are related to other services). The 30 features are subdivided into the three categories outlined in section 2, so that 4 features belong to the "intrinsic" category, 19 features to the "traffic" category and 7 features to the "content" category. For more details about the feature extraction process the reader is referred to [14]. Feature values have been linearly normalized so that their value is always within the range 0,..,1.

A training set made up of 122 normal samples, 6 U2R attacks, 539 R2L attacks, 1 probing and 57 DoS attacks corresponding to a total of 725 *ftp* connections has been extracted from the available data set. A test set made up of 7436 connections has been also extracted. In particular, it is made up of 5128 connections related to normal traffic and 2308 connections related to attacks, each one belonging to one out of the four attack classes. In particular 125 attack connections were related to attack types not included in the training set so that classification results related to these patterns allowed to test the capability of the pattern recognition approach to assign new attack types to the correct attack class.

At present, we solved the problem of unbalanced classes in the training set by augmenting the number of patterns related to poorly represented traffic types. In particular we carried out a number of preliminary classification experiments in order to identify which traffic types needed to be augmented in order to produce reliable performances on the test set. Consequently, we populated the training set by a number of copies of those connections so that the training set used in the experiments was made up of 157 normal samples, 46 U2R attacks, 598 R2L attacks, 25 probing and 57 DoS attacks corresponding to a total of 833 *ftp* connections. This heuristic approach is in agreement with the practical observation that if the same attack type is carried out a number of times, its connections exhibit quite similar feature values (i.e., these feature values correspond to the *signature* of that attack). Further work on the problem of unbalanced classes is obviously necessary.

Table 2 shows the performances on the test set of three neural networks trained using distinct feature representations, i.e., the 4 intrinsic features, the 7 content features, the 19 traffic features. In addition, the performances of a neural network trained using the entire 30 dimensional feature vector is reported for the sake of comparison. These networks are fully-connected multi-layer perceptrons (MLPs) with three layers of neurons. Each network has 5 output neurons (as the number of data classes), a number of inputs equal to the number of features and a hidden layer made up of 5 neurons for the networks trained on distinct feature representations, and 15 neurons for the network trained using all the 30 available features. Neural networks have been trained using the backprop algorithm, with different learning rates and different random starting weights. Reported results represent the best performances attained on the test set. Classification results are reported in terms of the overall classification error, the average classification cost computed according to the cost matrix shown in Table 1, and the false alarm rate. (Other researchers used the cost matrix shown in Table 1 to weight errors according to their severity [6]).

Table 1: Cost matrix used to evaluate the confusion matrix related to each classifier

		Assigned class				
		Normal	U2R	R2L	Probing	DoS
True class	Normal	0	2	2	1	2
	U2R	3	0	2	2	2
	R2L	4	2	0	2	2
	Probing	1	2	2	0	2
	DoS	2	2	2	1	0

The overall performances of neural networks, except for the network trained on traffic features, are quite similar each other, the network trained on the content features providing the best results. This result indicates that the content feature set is well suited for the type of traffic at hand, while the reverse is true with respect to the traffic feature set.

Table 2: Neural networks trained on distinct feature sets and on the entire feature set

Classifier Type	Overall Performances		
	% error	**Average cost**	**% false alarms**
MLP - 4 intrinsic features	1.51	0.031	3.19
MLP - 7 content features	1.20	0.024	2.25
MLP - 19 traffic features	9.83	0.200	23.94
MLP – 30 features	1.55	0.029	3.57

Table 3: Neural networks trained on distinct feature sets and on the entire feature set

Classifier Type	***Known* attacks**		***unknown* attacks**	
	% error	**Average cost**	**% error**	**Average cost**
MLP - 4 intrinsic features	0.73	0.017	16.00	0.320
MLP - 7 content features	0.82	0.017	14.40	0.280
MLP - 19 traffic features	2.11	0.047	31.20	0.880
MLP – 30 features	0.64	0.011	12.00	0.192

Table 3 reports the performances of the four neural networks on known attacks, i.e., attack types included in the training set, and unknown attacks, i.e., the 125 attack samples related to attack types not included in the training set (each unknown attack type belongs to one of the four attack classes). While performances of the network trained on the traffic features exhibit high error rates and high costs, in agreement with its overall performances, the two networks trained on the intrinsic and content features exhibit a quite similar behaviour. In particular, they exhibit a good generalization capability with respect to unknown attacks. However, as it can be expected, the best results in terms of attack detection is provided by the network trained on the overall feature set.

According to these results, two multiple classifier ensembles have been considered. The first one is made up of the three classifiers trained on the three distinct feature sets. Results are shown in Tables 4 and 5. Combination allows the improvement of the overall performances with respect to the individual classifiers, as well as with respect to the neural network trained on the 30-dimensional feature vector. With respect to attack detection the best performances are attained by the neural network trained on the 30-dimensional feature vector, especially if the capability of detecting *unknown* attacks is considered. Thus combining does not improve the generalization capabilities while improves overall performances. However, since the overall cost takes into account he trade-off between attack detection capabilities and false positives, thus it can be concluded that the multiple classifier approach based on distinct feature representation seems to be the more suited solution for implementing an IDS

Table 4: Combination of three neural networks trained on three distinct feature sets

Combination Technique	**Overall Performances**		
	% error	**Average cost**	**% false alarms**
Majority	0.89	0.018	1.29
Bayesian Average	0.87	0.017	1.33
Belief	0.82	0.014	0.87

Table 5: Combination of three neural networks trained on three distinct feature sets

Combination Technique	***known* attacks**		***unknown* attacks**	
	% error	**Average cost**	**% error**	**Average cost**
Majority	0.64	0.014	17.60	0.344
Bayesian Average	0.60	0.013	16.80	0.296
Belief	0.87	0.015	17.60	0.248

6 Conclusions

Researchers have recently proposed to use pattern recognition techniques to provide IDSs with generalization capabilities. While the reported results show the effectiveness of these pattern recognition approaches, the generalization capabilities of pattern recognition tools may originate high false alarm rates, as it is very difficult to collect a representative set of normal traffic samples [3].

The reported results showed the effectiveness of ensemble learning approaches in providing more reliable results, as the final decision depends on the agreement among distinct classifiers. In particular better results have been obtained by the fusion rule based on the "belief" function paradigm because it takes into account the different discriminative power provided by the considered feature sets. Other combination schemes should be devised to further improve the presented figures. In addition, more extensive testing is required to compare IDSs based on pattern recognition tools with traditional IDSs.

With respect to the capability of ensemble learning approaches of providing a better trade-off between generalization capabilities and false alarm rate, it can be concluded that combination reduces the overall error rate, but may also reduce the generalization capabilities. This aspect should be further investigated in order to deploy effective IDSs based on pattern recognition.

References

[1] J. McHugh, A. Christie, and J. Allen, "Defending Yourself: The Role of Intrusion Detection Systems", IEEE Software, Sept./Oct. 2000, pp. 42-51

[2] P.E. Proctor, The Practical Intrusion Detection Handbook, Prentice Hall, 2001

[3] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, E. Storner, State of the Practice of Intrusion Detection Technologies, (Tech. Rep. CMU/SEI-99-TR-028, 2000)

[4] R. Duda, P. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2001

[5] H. Debar, M. Becker, D. Siboni, "A Neural Network Component for an Intrusion Detection System", Proc. of the IEEE Symp. on Research in Security and Privacy, Oakland, CA, USA, 1992, pp. 240-250.

[6] J. Ryan, M.J. Lin, R. Miikkulainen, "Intrusion Detection with Neural Networks", in: Advances in Neural Information Processing Systems 10, M. Jordan et al., Eds., Cambridge, MA: MIT Press, 1998 pp. 943-949.

[7] S.C. Lee, D.V. Heinbuch, "Training a Neural-Network Based Intrusion Detector to Recognize Novel Attacks", IEEE Trans. on Systems, Man, and Cybernetics, Part A, 31, 2001, pp. 294-299.

[8] A.K. Ghosh and A. Schwartzbard, “A Study in Using Neural Networks for Anomaly and Mis use Detection", Proc. of the USENIX Security Symposium, August 23-26, 1999, Washington, USA.

[9] J. Cannady, “An adaptive neural network approach to intrusion detection and response”, PhD Thesis, School of Comp. and Inf. Sci., Nova Southeastern University, 2000.

[10] J.M. Bonifacio et al. “Neural Networks applied in intrusion detection systems”, Proc. of the IEEE World congress on Comp. Intell. (WCCI ’98), 1998.

[11] C. Elkan, “Results of the KDD’99 Classifier Learning”, ACM SIGKDD Explorations, 1, 2000, 63-64.

[12] J. Kittler and F. Roli (eds.), Multiple Classifier Systems, LNCS 2096, Springer, 2001.

[13] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, “On Combining Classifiers”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(3), 1998, pp. 226-229.

[14] W. Lee and S.J. Stolfo, “A framework for constructing features and models for intrusion detection systems”, ACM Trans. on Inform. and System Security, 3(4), 2000, 227-261.

[15] T.G. Dietterich, “Ensemble Methods in Machine Learning”, Proc. of the 1st Intern. Workshop on Multiple Classifier Systems, Cagliari, Italy, June 2000, LNCS 1857, Springer, 2000, pp. 1-15.

[16] Xu L., A. Krzyzak and C.Y. Suen (1992). Methods for combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. Systems, Man and Cybernetics 22, 418-435.

[17] http://www.ll.mit.edu/IST/ideval

[18] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html