

Classifying Attacks in a Network Intrusion Detection System Based on Artificial Neural Networks

Mohammad Reza Norouzian*, Sobhan Merati**

*Information Technology Department, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran

** Information Technology Department, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran

m_norouzian@iasbs.ac.ir, s_merati@iasbs.ac.ir

Abstract—Nowadays with the dramatic growth in communication and computer networks, security has become a critical subject for computer systems. A good way to detect the illegal users is to monitoring these user's packets. Different algorithms, methods and applications are created and implemented to solve the problem of detecting the attacks in intrusion detection systems. Most methods detect attacks and categorize in two groups, normal or threat. This paper presents a new approach of intrusion detection system based on neural network. In this paper, we have a Multi Layer Perceptron (MLP) is used for intrusion detection system.

The results show that our implemented and designed system detects the attacks and classify them in 6 groups with the approximately 90.78% accuracy with the two hidden layers of neurons in the neural network.

Keywords—Artificial Neural Networks; Intrusion Detection System; Multilayer Perceptron; Network Security.

I. INTRODUCTION

Today because of existence of Internet and highly increasing in using computers and Internet by people, companies and governments doing their works and the dependency of systems to computer networks, sustain the security plays an essential role to prevent the attacks.

The highly connected computing world has also equipped the intruders and hackers with new facilities for their destructive purposes.

This article mainly discusses the application of the artificial neural networks in the field of IDS research. We have designed a network intrusion detection system based on the artificial neural networks using Multi Layer Perceptron (MLP); and the testing results of the prototype system proved the validity of the method and the advantages over other methods suggested. In many previous studies [1], [2], [3] the implemented system is a neural network with the capability of detecting normal or attack connections, in the present study a more general problem is considered in which the attack type is also detected. This feature enables the system to suggest proper actions against possible attacks. The promising results of the present study show the potential applicability of ANNs for developing practical IDSs.

A. Intrusion Detection Systems

The timely and proper detection of computer and network system intrusions has always been an important for

system administrators and information security researchers. While the complexities of host computers already made intrusion detection a difficult attempt, the increasing in spreading distributed network-based systems and insecure networks such as the Internet has greatly increased the need for intrusion detection [4], [5].

Most current approaches to the process of detecting intrusions utilize some form of rule-based analysis. Rule-based analysis sets of predefined rules that are provided by an administrator or created by the system. Expert systems are the most common form of rule-based intrusion detection approaches. The use of expert system techniques in intrusion detection mechanisms was a significant milestone in the development of effective and practical detection-based information security systems. An expert system consists of a set of rules that encode the knowledge of a human "expert". These rules are used by the system to make conclusions about the security-related data from the intrusion detection system. Unfortunately, expert systems require frequent updates to remain current. This design approach usually results in an inflexible detection system that is unable to detect an attack if the sequence of events is even slightly different from the predefined profile. While increasing the level of abstraction of the rule-base does provide a partial solution to this weakness, it also reduces the granularity of the intrusion detection device. The problem may arise that the intruder or hacker is an intelligent and flexible agent while the rule-based IDSs obey fixed rules. In fact expert systems suffer from the updating, the searching and the matching of the rule sets.

B. Classification of IDS

A few number of taxonomies of IDS proposed by different researchers at different time have been describe in this section. Based on the taxonomy, a few numbers of IDSs, in the context of this work, have also been described.

1) *Signature based (Misuse) IDS*: This type of intrusion detection system contains a database of know vulnerabilities. It monitors traffic and seeks a pattern or a signature match. This means, it operates in much the same way as a virus scanner, by searching for a known identity or signature for each specific intrusion event. It can be placed on a network to watch the network vulnerabilities or can be placed on a host [6].

Signature-based IDS examine ongoing traffic, activity, transaction, or behaviour for matches with known patterns of even specific to known attacks and it raises alarm only when the so called match is found. This is why the number of false positive alarms is comparatively less in signature based IDS. If the system is not entirely new i.e. when it has an up to date database of signatures of known attacks, this technique works extremely well.

2) *Anomaly based IDS*: Also known as Heuristic or Behaviour based, Anomaly based IDS analyzes the traffic patterns and determine normal activities. After that, it applies statistical or heuristic measures to event to determine if they match with this normal behaviour. Events which do not match with the accepted normal behaviour patterns are considered as attacks [7].

By creating patterns of normal behaviour, anomaly-based IDS systems can observe when current behaviour deviates statistically from the normal. This capability theoretically gives anomaly-based IDSs abilities to detect new attacks that haven't been seen before or close variants to previously known attacks. It means, these types of IDS may identify any possible attacks [8].

Since normal behaviour can be changed over time, this type of system requires frequent retraining of the behaviour profile, lack of which results either in unavailability of the intrusion detection system or in additional false alarms. This type of systems normally provides comparatively high false alarms as something even slightly different from normal behaviour is considered as an attack by this types of IDS. It also requires expertise to figure out what triggered an alarm.

3) *Host based IDS (HIDS)*: Host Based IDS is installed locally on host machines. It works on information collected from within an individual computer system. It utilizes information sources like operating system audit trails, C2 audit logs and system logs. HIDS can be installed on different types of machines namely servers, workstations and notebook computers.

Host-based IDS can analyze activities on the host it monitors at a high level of detail. It can often determine which processes and/or users are involved in malicious activities. It can monitor events that are local to a host and can detect successful or failure of attacks that cannot be seen by a network-based IDS. Host-based IDSs can use host-based encryption services to examine encrypted traffic, data, storage and activity. They are unaffected by switched networks and independent of network topology. They can provide thorough information gathered via logs and audit; for example Kernel logs records that the user is.

4) *Network based IDS (NIDS)*: Network based IDS monitors the traffic on its entire network segment. This is generally accomplished by placing the network interface card in promiscuous mode to capture all network traffic segments. These network traffic packets are checked network by the IDS to find the attacks. Network based IDS can reassemble packets, look at headers, determine if there are any predefined patterns or signature match. Depending

on this pattern or signature matching, IDS decides about the attacks.

Network-based IDSs can monitor an entire, large network with only a few well-situated nodes or devices and impose little overhead on a network. This type of IDSs is mostly passive devices that monitor ongoing network activity without adding signature overhead or interfering with network operation. They can monitor network for malicious activity on known ports such as http port 80. They are easy to secure against attack and may even be undetectable to attackers; they also require little effort to install and use on existing networks. Furthermore, they are operating systems independents.

C. Artificial Neural Networks

Artificial neural network is an information processing model that is inspired by the biological nervous systems, such as brain, process information. It tries to represent the physical brain and thinking process through electronic circuit or software. Artificial neural network is the network of individual neurons. Each neuron is a neural network acts as an independent processing element. Each processing element (neuron) is fundamentally a summing element followed by an activation function. The output of each neuron (after applying the weight parameter associated with the connection) is fed as the input to all of the neurons in the next layer. Like human or other brain, neural networks also learn by example or training, they cannot define or program to perform a specific task. Neural networks perform very successfully for recognizing and matching complicated or incomplete patterns. The most successful application of neural network is classification or categorization and pattern recognition. The learning process is essentially an optimization process in which the parameters of the best set of connection coefficients (weights) for solving a problem are found and includes the following basic steps [9]:

- Present the neural network with a number of inputs
- Check how closely the actual output generated for a specific input matches the desired output.
- Change the neural network parameters to better approximate the outputs.

There are two different learning methods for the neural networks: supervised and unsupervised. In supervised learning method, the network learns the desired output for a given input or pattern. The well known architecture of supervised neural network is the Multi-Level Perceptron (MLP); the MLP is employed for Pattern Recognition problems. On the other hand, in unsupervised learning method, the network learns without specifying desired output. Self-Organizing Maps (SOM) are popular unsupervised training algorithms; a SOM tries to find a topological mapping from the input space to clusters. SOM are employed for classification problems.

1) *Neural Network Intrusion Detection Systems*: Some IDS designers utilize ANN (Artificial Neural Network) as a

pattern recognition technique. During training, the neural network parameters are optimized to associate outputs (each output represents a class of computer network connections, like normal and attack) with corresponding input patterns (every input pattern is represented by a feature vector extracted from the characteristics of the network connection record). When the neural network is used, it identifies the input pattern and tries to output the corresponding class [10]. The most commonly reported application of neural networks in IDSs is to train the neural net on a sequence of information units, each of which may be an audit record or a sequence of commands.

2) *False positive and False Negative*: Intrusion Detection systems are not perfect systems. So mistake may happen in them. Two considerable mistakes of intrusion detection systems are False Positive and False Negative. Normally, when the IDS detect an ongoing attack in the computer system, it raises alarm for taking action by the administrator. False Positive is a state where IDS raises alarm of attacks for something that is not really an attack. If too many false positive occur, it makes the administrator less confident about the alarms and hence, there is a possibility of ignoring the real attack by the administrator. On the other hand, False Negative is state where IDS does not raise an alarm for a real attack. This means, a real attack is passing by without the action being taken. This must be very dangerous for the system as the real attacks are completely being overlooked.

II. SYSTEM ARCHITECTURE

The main merit of the artificial neural networks includes the ability of faster information processing, the ability of

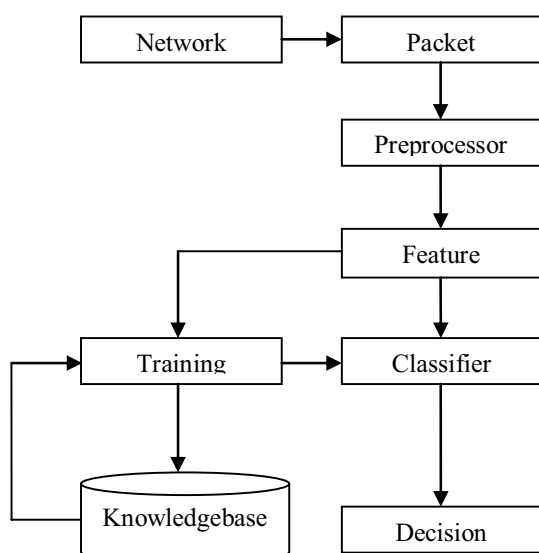


Figure1. Block diagram of system overview

classification and the ability of learning and self-organization. Because of these abilities of the artificial neural networks, the network intrusion detection system can analyze the network captured packets and detect whether it would be an intrusion or not. From the view of architecture, the diagram of system includes several modules, which has shown in Figure 1.

Based on the figure above, there are several modules that introduce the network intrusion detection system based on the artificial neural networks architecture. Now we are about to describe this layering structure step by step.

- **Packet Monitor**: This module monitors network stream real time and capture packets to serve for the data source of the NIDS.

- **Preprocessor**: In preprocessing phase, network traffic collected and processed for use as input to the system.

- **Feature Extractor**: This module extracts feature vector from the network packets (connection records) and submits the feature vector to the classifier module.

- **Classifier**: The function of this module is to analyze the network stream and to draw a conclusion whether intrusion happens or not.

- **Decision**: When detecting that intrusion happens, this module will send a warning message to the user.

- **Knowledgebase**: This module serves for the training samples of the classifier phase. As you know, the artificial neural networks can work effectively only when it has been trained correctly and sufficiently. The intrusion samples can be perfected under user participation, so the capability of the detection can improve continually.

All of these modules together make the NIDS architecture system based on the artificial neural networks, and the function of each module has been introduced briefly. The present study is aimed to solve a multi class problem in which not only the attack records are distinguished from normal ones, but also the attack type is identified.

III. DATASET

1) The dataset Network Intrusion Detection was chosen from The UCI (University of California, Irvine) KDD (Knowledge Discovery in Databases) Archive [11]. The dataset is the collection of network related information that was captured over a period of time. The dataset downloaded from The UCI KDD Archive consists of 41 attributes, 300,000 instances and 38 different classes. A subset of the data (approximately 10%) which contained the desired attack types and a reasonable number of normal events were selected manually.

B. Attack Types

There are at least four different known categories of computer attacks including denial of service (DOS) attacks, user to root (U2R) attacks, remote to user (R2L) attacks and probing attacks [12]. In this study six types of attacks are considered: Smurf, Teardrop, Satan, Guest, Warezclient and Buffer overflow. These six attack types were selected from

four different attack categories (denial of service, probing, user to root and remote to user) to check for the ability of the intrusion detection system to identify attacks from different categories. Availability of enough data records was the other factor in choosing these six specific types. In the following paragraphs, a description of the attack types is provided.

1) *Smurf*: In this attack, attackers use ICMP echo request packets directed to IP broadcast addresses from remote locations to create a denial-of-service attack. The attacker sends ICMP “echo request” packets to the broadcast address (xxx.xxx.xxx.255) of many subnets with the source address spoofed to be that of the intended victim. Any machines that are listening on these subnets will respond by sending ICMP “echo reply” packets to the victim. The Smurf attack is effective because the attacker is able to use broadcast addresses to amplify what would otherwise be a rather innocuous ping flood.

2) *Teardrop*: A Teardrop is a denial of service which involves sending mangled IP fragments with overlapping, over-sized payloads to the target machine. This can crash various operating systems due to a bug in their TCP/IP fragmentation re-assembly code.

3) *Satan*: is a probing intrusion which automatically scans a network of computers to gather information or find known vulnerabilities. The network probes are quite useful for attackers planning a future attack [13].

4) *Guest*: The Guest attack is a variant of the Dictionary attack that the dictionary attack is a Remote to Local User attack in which an attacker tries to gain access to some machine by making repeated guesses at possible usernames and passwords. On badly configured systems, guest accounts are often left with no password or with an easy to guess password. Because most operating systems ship with the guest account activated by default, this is one of the first and simplest vulnerabilities an attacker will attempt to exploit.

5) *Warezclient*: Warezclient attack can be launched by any legal user during an FTP connection after warezmaster attack has been executed. During warezclient attack, users download the illegal “warez” software that was posted earlier through a successful warezmaster attack.

6) *Buffer overflow*: Buffer overflows are of particular interest as they are potentially exploitable by malicious users, and have historically accounted for a significant percentage of the software vulnerabilities that are published each year buffer overflow vulnerability occurs when data can be written outside the memory allocated for a buffer, either past the end or before the beginning.

Table I shows detailed information about the number six attack records from the 494021 records in the 10% subset dataset.

C. Feature Extraction

Feature selection and extraction is one of the important problems in implementing the intrusion detection system. Network stream itself is not suitable directly as the input of the Classifier module, so it is necessary to extract some features from the network stream. The features extracted from the network stream are a feature vector which serves for the description of the packet. Whether the feature vector can describe the network stream correctly and efficiently or not has a large effect on the efficiency and correctness of the NIDS.

TABLE I. NAMES, TYPES AND NUMBERS OF RECORDS IN 10% OF DATASET

Attacks Type	#Samples	Category
Smurf	280790	DOS
Normal	97277	Normal
Teardrop	979	DOS
Buffer overflow	30	U2R
Guest	53	R2L
Warezclient	1020	R2L
Satan	1589	Probe

Based on the UCI KDD Archive [14] dataset each record (connection) is described with 41 features. 22 of these features describe the connection itself and 19 of them describe the properties of connections to the same host in last two seconds. In many attack scenarios, the signature of the attack record is identified through examination of some features in a sequence of records. Therefore, the IDS should analyze the service types used by the same user in previous connections and for this purpose these 19 features describing past events in the computer network were included in the feature vector.

A complete description of all 41 features is available [3], [15]. Instead of describing all the features, here for making data as an input in training phase we should select features first. So we perform some examinations and based on the number of features that have most frequently occurring in attacks types, we considered to select 13 features such as: *source bytes*, *diff srv rate*, *destination byte*, *error rate*, *srv different host rate*, *flag*, *land*, *service*, *#root*, *srv error rate*, *protocol type* and *count* to use in our intrusion detection.

For example, for the protocol type the possible numerical values were: tcp=0, udp=1, icmp=2. This numerical representation was necessary because the feature vector fed to the input of the neural network has to be numerical.

The ranges of the features were different and this made them incomparable. Some of the features had binary values where some others had a continuous numerical range. As a result, the features were normalized by mapping all the different values for each feature to [0, 1] range.

D. Classifier module

Classifier module is the most widely used neural network model, which has made a good figure in the field of Pattern Recognition. In fact, the problem to be solved by the intrusion detection system is to differentiate "Good Packets" from "Bad Packets" in the network stream. In this sense, "Intrusion Detection" can be understood as "Pattern Recognition" to some extent. From this point of view, we select Classifier module in the design of the NIDS prototype based on the artificial neural networks. There are several problems which should be considered in the design of the Classifier:

- The number of the hidden layer. In general application, one hidden layer is enough, but we set two hidden layers in the Classifier module (at the end of paper it has indicated why it is considered), consequently we have a three layers neural network. (The input layer is not counted because it acts just like a buffer and no processing takes place in it; however, the output layer is counted)

- The dimension of the input layer and the output layer. Generally speaking, the dimension of the input layer is the number of the features selected, and the dimension of the output layer is the number of sorts that can be classified by the Classifier.

- The transfer function. In general, the function $\text{Logsig}(x) = 1 / (1 + \exp(-x))$ can be used in the Classifier model.

- The learning function. *trainGD* function can be used in the Classifier which works based on Back-propagation algorithm.

- Initialization of the weight and the threshold. The weight of different neural units and the threshold will be set to the small random number from 0 to 1 and as to threshold; we assign the output vales under 0.5 to zero and upper 0.5 to one. By doing so, the convergence of the BP network can be ensured. After taking these problems into account carefully, we designed a Classifier module.

MATLAB Neural Network Toolbox was used for the implementation of the MLP networks. Using this tool one can define specifications like number of layers, number of neurons in each layer, activation functions of neurons in different layers, and number of training epochs. Then the training feature vectors and the corresponding desired outputs can be fed to the neural network to begin training.

IV. TESTING RESULT

The implemented intrusion detector was a three layer MLP. In this database 381738 samples are exist for evaluating in testing and training phase, 2000 of which are used for training, and the 200 are used for testing.

Table II gave the testing results for the seen data (train set) in intrusion detection system and relaying on the results classification rate (alarm rate) on training set confirmed that the classification really close to 100%.

TABLE II. TESTING SENN DATA

Attacks Types	Alarm rate%	False Negative rate%	False Positive rate%
Smurf	98.62	0.83	0.55
Teardrop	97.8	1.34	0.86
Satan	98.1	1.18	0.72
Guest	96.91	2	1.09
Buffer Overflow	96.38	2.17	1.45
Warezcilent	97.94	1.17	0.89

Table III gave the testing results for the unseen data (test set) in intrusion detection system. From the Table 3, we can see that the Classifier has the ability to detect the unseen intrusions, that is to say, the Classifier can detect the unseen intrusions. This is mainly because that the artificial neural networks have the ability of classification and self-organization. For these six kinds of attacks, the testing results are satisfactory.

TABLE III. TESTING UNSENN DATA

Attacks Types	Alarm rate%	False Negative rate%	False Positive rate%
Smurf	94.1	3.54	2.36
Teardrop	90.72	5.66	3.62
Satan	94.44	3.45	2.11
Guest	91.36	5.62	2.84
Buffer Overflow	84.69	9.18	6.13
Warezcilent	89.4	6.04	4.56

To sum up, on the detection of the seen intrusions, the prototype system has a better performance with high correctly detected rate and a low false alarm rate.

A. Impact of Layers in Neural Network

In this study MLP neural networks was implemented and in the training back-propagation algorithm was used. However the back-propagation algorithm becomes more complicated and consumes more memory and time when a hidden layer is added to the neural network. Furthermore, the resulting neural network is more complicated and less memory efficient. It seems that with increasing to the layers of this system outputs will get better improved but after rising the layer in neural network we can see that with growing in number of layers it can just increase the overhead in system and consumes more memory and time without any satisfactory result in which improve the performance of the IDS. Figure 2 shows that the MSE (mean square error) in three layer prototype will be at the lowering level. Therefore, it is always desirable to solve the problem with a simpler classifier.

B. Discussion

Actually there were three categories of incorrect outputs: false positive, false negative, and irrelevant neural network output. The irrelevant outputs were those that did not represent any of the output classes in the data set. While in a two state neural network implemented with one output neuron there is no irrelevant output state. An analysis

showed that in the three layer neural network 90.78% correct classification was happened and we considered the irrelevant outputs in false negative category.

Are the results presented is satisfactory? To answer this question, they should be compared to the results of similar studies. In a previous study [1], a result of more than 99% correct classification on this dataset using the neural network structure {41-40-40-1} was reported. However, a two class problem was solved in which the records were classified either in normal or in attack classes. . In another similar study with different dataset [1], the success rate was comparable to the results of the present study and again a two class problem was implemented.

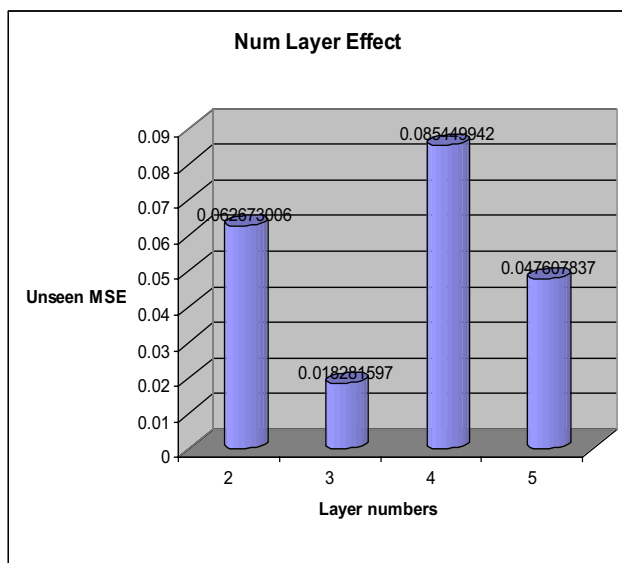


Figure1. Impact of layers in Unseen MSE

V. CONCLUSIONS AND FUTURE WORK

An approach for a neural network based intrusion detection system, intended to classify the normal and attack patterns and the type of the attack, has been described in this paper. It should be mentioned that the long training time of the neural network was mostly due to the huge number of training vectors of computation facilities. However, when the neural network parameters were determined by training, classification of a single record was done in a negligible time. Therefore, the neural network based IDS can operate as an online classifier for the attack types that it has been trained for.

Although the classification results were slightly better in the three layer network, application of a less complicated neural network was more computationally and memory wise efficient.

From the practical point of view, the experimental results imply that there is more to do in the field of artificial neural network based intrusion detection systems especially solving irrelevant outputs. The implemented system solved

classification problem. However, its further development to several classes is straightforward. As a possible future development to the present study, one can include more attack scenarios in the dataset. Practical IDSs should include several attack types. In order to avoid unreasonable complexity in the neural network, an initial classification of the connection records to normal and general categories of attacks can be the first step. The records in each category of intrusions can then be further classified to the attack types.

REFERENCES

- [1] James Cannady, "Artificial neural networks for misuse detection," Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, 1998.
- [2] J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion Detection with Neural Networks," AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop, Providence, RI, pp. 72-79, 1997.
- [3] Srinivas Mukkamala, "Intrusion detection using neural networks and support vector machine," Proceedings of the 2002 IEEE International Honolulu, HI, 2002.
- [4] Mukherjee, B., Heberlein, L.T., Levitt, K.N, "Network Intrusion Detection". IEEE Network. pp. 28-42, 1994.
- [5] Kabiri P, Ghorbani A, "A. Research in intrusion detection and response – a survey". International Journal of Network Security, 2005.
- [6] Helman, P., Liepins, G., and Richards, "Foundations of Intrusion Detection". In Proceedings of the Fifth Computer Security Foundations Workshop pp. 114-120, 1992.
- [7] Anderson, D., Frivold, T. & Valdes, "Next-generation Intrusion Detection Expert System (NIDES): A Summary". SRI International Technical Report SRI-CSL-95-07, 1995.
- [8] P. Garcí'a-Teodoroa, J. Dí'az-Verdejoa, G. Macía'-Ferna'ndeza, E. Va'zquezb, "Anomaly-based network intrusion detection: Techniques, systems and challenges", Elsevier, 2009.
- [9] Sergios Theodorios and Konstantinos Koutroumbas, "Pattern Recognition", Cambridge: Academic Press, 1999.
- [10] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, pp. 240 – 250, 1992.
- [11] Archive, U. K. "KDD Cup 1999 Data." Retrieved April 20, 2009, from <http://www.sigkdd.org/kddcup/index.php>
- [12] R. Lippmann, JW Haines, DJ Fried, J Korba, K Das, "The 1999 DARPA off-line intrusion detection evaluation" – Elsevier, 2000.
- [13] Kristopher Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems," Masters Thesis, MIT, 1999.
- [14] Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis "Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection, Results from the JAM Project", pp. 20-27, 1999.
- [15] <http://www.sigkdd.org/kddcup/index.php?section=1999&method=tas> k