7

Intrusion Detection System and Artificial Intelligent

Khattab M. Alheeti
Alanbar University
Iraq

1. Introduction

In this chapter we will look to the essential demonstrates the basic concepts of Intrusion Detection System and explain the role of artificial intelligence in the promotion and development of Intrusion Detection System using artificial intelligence. We will then explain the model the new intelligent, who has much on the distinction between regular normal connection and unusual or abnormal connection, Use of neural networks with data normal with out Fuzzification to build a new predicative model, and we have another predicative model but the data that is entered into by the Fuzzification data. And what follows mention of the topics that will take them in our chapter.

On completing this chapter, you will be able to:

- Explain the major categorized and classification of the IDSs.
- Describe network-based IDSs.
- Describe host-based IDSs.
- Explain how IDS management communication works.
- Explain the principles of artificial intelligence.
- Describe the type of the neural network.
- Explain Crisp set and Fuzzy set.
- Describe the predicative model with normal data
- Describe the predicative model with Fuzzification data.
- Conclusion.
- Future works.
- References.

2. Overview of intrusion detection system

An intrusion can be defined as ‘’an act of a person of proxy attempting to break into or misuse a system in violation of an established policy’’ [Malik 2002]. So to protect systems from intruders, intrusion detection system is needed. IDS is software and/or hardware system for monitoring and detecting data traffic or user behavior to identify attempts of illegitimate accessing system manipulation through a network by malware and/or intruders (crackers, or disgruntled employees). ID has been used to protect information systems along with prevention-based mechanisms such as authentication and access control. An ID cannot directly detect attacks within properly encrypted traffic.

Intrusion detection systems can be classified as *network-based* and *host-based* according to the information source of the detection. Network-based IDS monitors the network traffic and looks for network-based attacks, while host-based IDS is installed on host and monitors the host audit trail. Intrusion detection systems can be roughly classified as *anomaly detection* and *misuse detection*. Anomaly detection is based on the normal behavior of a subject (e.g., a user or a system). Any action that significantly deviates from the normal behavior is considered intrusive. Misuse detection is based on the characteristics of known attacks or system vulnerabilities, which are also called *signatures*. Any action that matches the signature is considered intrusive. Both anomaly detection and misuse detection have their limitations.

Misuse-base detection detects attacks based on signatures (known attacks signatures), at which the traffic pattern compared with these signatures, if a match is found, then it is reported as an attack, otherwise it is not. So misuse detection cannot detect novel attacks. On the other hand, anomaly-based detection depends on monitoring system activity and classifying it as either normal or anomalous. The classification is based on heuristics or rules, rather than patterns or signatures, and will detect any type of misuse that falls out of normal system behavior.

The strength of the anomaly detection approach is that prior knowledge of the security flaws of the target systems is not required. Thus, it is able to detect not only known intrusion but also unknown intrusion. In addition, this approach can detect the intrusion that is achieved by the abuse of legitimate users or masqueraders without breaking security policy [Denning 1987, Porras 1992]. However, it has several limitations, such as high false positive detection error, the difficulty of handling gradual misbehavior and expensive computation [Mykerjee and et al 1994]. In contrast, the misuse detection approach detects only previously known intrusion signatures. The advantage of this approach is that it rarely fails to detect previously notified intrusions [Denning 1987]. However, this approach cannot detect new intrusions that have never previously been monitored. Furthermore, this approach is known to have other drawbacks such as the inflexibility of misuse signature rules and the difficulty of creating and updating intrusion signature rules [Porras 1992, Kumar 1995]. These strengths and limitations of the two approaches imply that effective IDS should employ an anomaly detector and a misuse detector in parallel [Mykerjee and et al 1994]. However, most available commercial IDS's use only misuse detection because most developed anomaly detector still cannot overcome the limitations described above. This trend motivates many research efforts to build anomaly detectors for the purpose of ID [Kim 2002]. However, the nature of current and future threats in conjunction with ever larger Information Technologies (IT) system systems urgently requires the development of automated and adaptive defensive tools.

2.1 Intrusion Detection System (IDS)

To understand what is ID, the meaning of intrusion should be declared. An intrusion can be defined as [Kumar 1995, Bace & Peter 2001]: "Any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource". Intrusion detection becomes essential security mechanism to protect systems and networks. It attempts to detect improper or inconsistent activity in a computer network, or on a host, by the exploration of certain kinds of data through monitoring. Such activities may be initiated from external intruder or from internal misuse. According to the monitored system, IDS could be categorized into [Song 2007, Sundaram 2002, Crothers 2003, and Kazienko & Piotr 2004]:

- Network-based IDS: is an independent platform that monitors the network backbones and look for attack scenarios by analyzing, examining, and monitoring network traffic data. Network Intrusion Detection Systems (NIDS) gain access to network traffic by connecting to a hub, network switch configured for port mirroring, or network tap. The NIDS reassemble and analyze all network packets that reach the network interface card. They do not only deal with packets going to a specific host – since all the machines in a network segment benefit from the protection of the NIDS. Network-based IDS can also be installed on active network elements, for example on router.
- Host-based IDS: reside on a particular computer and tries to detect malicious activity and provide protection for a specific computer system by monitoring the operating and file systems for signs of intrusion. This can be done through an agent placed on that host to identify intrusions by analyzing system calls, application logs, file-system modifications (binaries, password files) and other host activities and state.
- Hybrid of HIDS and NIDS, Host agent data is combined with network information to form a comprehensive view of the network. the main reason for introducing such hybrid IDS is the need to work online with encrypted networks and their data destined to the single host (only the source and destination can see decrypted network traffic).

The fact is that intrusion detection systems are not organs of what regulations deter more control and monitor, alarm any detect process will determine the intruder and where breach occurred, when and what the response depends on the design of the system. In addition, the alarm does not provide system security itself; it only to indicate that some sort of potentially malicious activity is being attempted.

2.2 Classification of IDS

ID is a network security tool that concerned with the detection of illegitimate actions. This network security tool uses one of two main techniques [Song 2007, Crothers 2003, and Kazienko & Piotr 2004]:

- Anomaly detection explores issues in ID associated with deviations from normal system or user behavior. It is based on the assumption that the characteristics of attacks are significantly different from normal behavior. Anomaly detection is capable of detecting unknown attacks or variants of known attacks if such attacks significantly change the monitored characteristics of the system. And deviations of normal usage of programs regardless of whether the source is a privileged internal user or an unauthorized external user. The disadvantage of the anomaly detections approach is that well-known attacks may not be detected, particularly if they fit the established profile of the user. Another drawback of many anomaly detection approaches is that a malicious user who knows that he or she is begin profiled can change the profile slowly over time to essentially train the anomaly detection system to learn the attacker's malicious behavior as normal.
- The second employs Misuse (Signature detection) refers to known attacks that exploit the known vulnerabilities of the system to discriminate between anomaly or attack patterns (signatures) and known ID signatures. The main disadvantage of misuse detection approaches is that they will detect only the attacks for which they are trained to detect (i.e. not capable of detecting novel or unknown attacks).

The IDS can operate as standalone, centralized application or integrated applications that create a distributed system. One may categorize IDSs in terms of behavior i.e., they may be Passive (those that simply generate alerts and log network packets). They may also be active which means that they detect and respond to attacks, attempt to patch software holes before getting hacked or act proactively by logging out potential intruders, or blocking services.
IDSs can run on either a continuous or periodic feed of information (Real-Time IDS and Interval-base IDS respectively) and hence they use two different ID approaches. Audit trail analysis is the prevalent method used by periodically operated systems. In contrast, the IDS deployable in real-time environments are designed for online monitoring and analyzing system events and user actions.
With on the fly processing, an ID performs verification of system events. Generally, a stream of network packets is constantly monitored. With this type of processing, ID uses the knowledge of current activities over the network to sense possible attack attempts (it does not look for successful attacks in the past). Figure (1.2) shows the classification of IDS from different point of views:

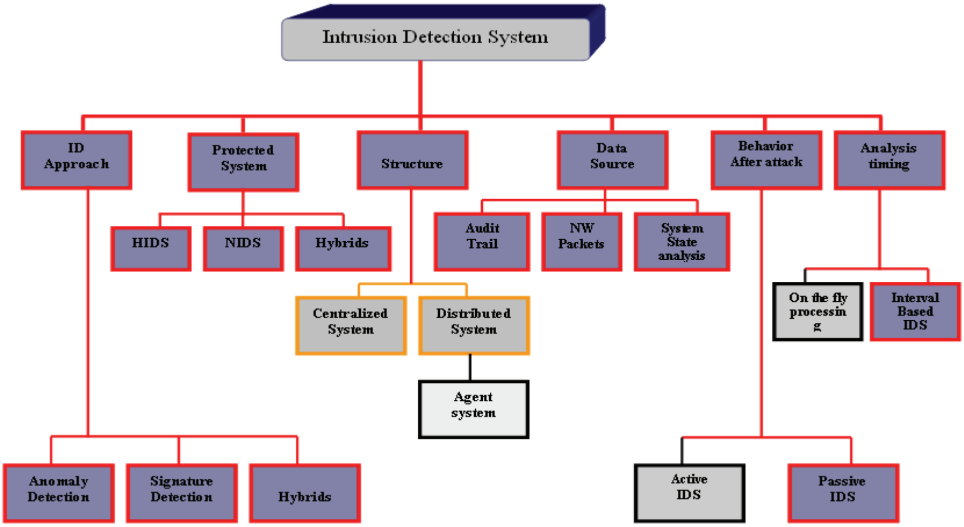


Fig. 1.2. The classification of IDS from six different points of views

3. Problem definition

Recently, the problem of computer systems intrusions grows and intruders become intelligent and flexible agents. The reason is that, new automated hacking tools appear every day, and these tools, along with various system vulnerability information, are easily available on the web. This problem can be solved by using appropriate software which is designed to detect and prevent intrusions.
Intrusion detection (ID) is an appealing concept since the current techniques used in computer security are not able to cope with dynamic and increasingly complex nature of computer systems and their security. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between “bad” connection,

called intrusion or attack, and "good" normal connections depending on special attributes (*features*) that are collected from *the packet header* and *audit trail* files (behavior during the connection). Such classifiers could be built using different approaches (statistical approaches, genetic algorithms, fuzzy systems, or neural networks).

To evaluate any intrusion detection system, dataset collected by Defense Advanced Research Project Agency is used. This dataset is a version of the 1999 DARPA intrusion detection evaluation data set prepared and managed by MIT Lincoln Labs. In this data set, 41 attributes that describe the different features of the corresponding connection (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). The value of the connection is labeled either as an attack with one specific packet header and the data payload.

There are 39 different attack types presented and falls exactly into one of the following four categories [Salvatore and et al 2000]:

1. Probing (surveillance and other probing): Probing is a class of attack where an attacker scans a network to gather information or find known vulnerabilities.
2. DOS (denial-of-service): it is a class of attack where an attacker makes a computing or memory resource too busy or too full handles legitimate requests thus denying legitimate users access to a machine.
3. U2R (User to root): unauthorized access to local super user (root) privileges exploits. It is a class of attacks where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.
4. R2L (A remote to local): unauthorized access from a remote machine. It is a class of attack where an attacker sends packets to a machine over a network, then exploits the machine's vulnerability to illegally gain local access as a user.

Another problem, current intrusion detection systems (IDS) examine all data features to detect intrusion or misuse patterns, use all the feature adds extra burden on the system, and extra features can increase computation time, and can impact the accuracy of IDS. Therefore, it is important to reduce number of features (attributes) that would be used to train the classifier. One of the major problems is to select the proper attributes (from the total 41 attribute in dataset) that have the best discrimination ability (between normal and attack connections). It is also important to choose the suitable classifier with high classification rate.

4. Objectives

One way to detect illegitimate use in network system is through monitoring unusual user activity by using IDS. Different methods used to build intrusion detection system (such as statistical, genetic, fuzzy genetic, neural networks etc). This chapter aims to study the classification ability of feed-forward neural network with actual data and feed-forward neural network with fuzzified data and compare their results from distinguishing accuracy point of view. Also, try to reduce the 41 attributes since some of these features will not affect the classification accuracy (or even may negatively affect it) at which their values are the same in different attack types or normal one. The main goal of this chapter is to improve classification rate of the discrimination ability (i.e. discriminate attacks from normal behavior).

In additional, most of the previous studies focused on classification of records in one of the two general classes-normal and attack, this chapter aim's to solve a multi-class problem at which the type of attack is also detected by the suggested intrusion detector. Using the reduced data sets, 5-class classifier is built (normal data belongs to class 5, probe belongs to class 1, denial of service belongs to class 2, user to super user belongs to class 3, remote to local belongs to class 4). The dataset is partitioned into 3 different parts (validation part, training part, and testing part). The evaluation of the system accuracy will depend on the testing results.

5. Artificial Neural Network

The idea of Artificial Neural Network (ANN) came from the idea of working human brain; the first step toward artificial neural networks came in 1943 when Warren McCulloch, a neurophysiologist, and a young mathematician, Walter Pitts, wrote a paper on how neurons might work. Think scientists in a way which can simulate the process, which occur in the human mind, and came to the knowledge of Neural Network, which falls under science artificial intelligence, so as to make computers intelligent devices, they can gain knowledge of the same way that acquires the rights of knowledge, they control the way weights during the learning. In addition, on the structural side large number of highly interconnected processing elements (neurons) working together. The neuron is the basic information processing unit of a Neural Network (NN); it consists of: A set of links, describing the neuron inputs, with weights W1, W2, …,Wm, An adder function (linear combiner) for computing the weighted sum of the inputs (real numbers):

$$U_j = \sum_{i=1}^{p} W j_i x_i \qquad (1)$$

And an activation function (squashing function) for limiting the amplitude of the neuron output.

$$\text{Tan-Sigmoid function} = 2/\ (1+\exp\ (-2*n))-1 \qquad (2)$$

In addition, there is extra weight value considered which is corresponding to the constant bias (extra input). The bias is an external parameter of the neuron; it can be modeled by adding an extra input. Figure (1.3) shows the neuron and bias, while Figure (1.4) shows the biological neuron:

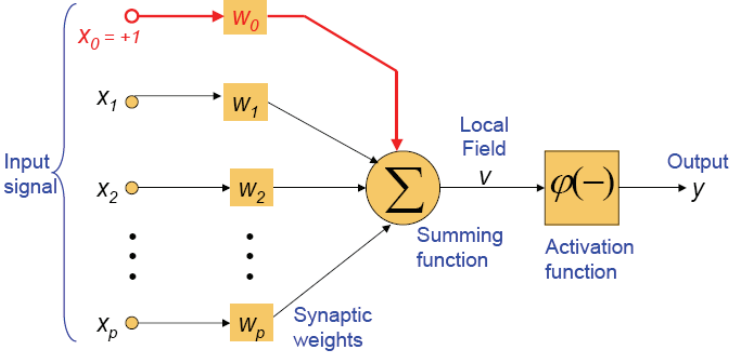


Fig. 1.3. The artificial neuron

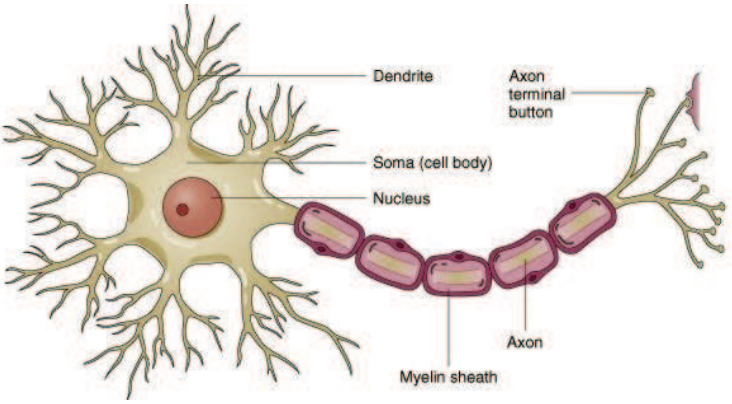


Fig. 1.4. Biological neuron

5.1 Classification of Neural Network

Neural networks can be classified into dynamic and static categories. Static (Feed-forward) networks have no feedback elements and contain no delays; the output is calculated directly from the input through feed-forward connections. The training of static networks was discussed in Backpropagation. In dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network. You saw some linear dynamic networks in Linear Filters.

Dynamic networks can also be divided into two categories: those that have only feed-forward connections, and those that have feedback, or recurrent, connections.

Generally classified neural networks on the basis of either training (learning) or architectures. There are two approaches to training-supervised and unsupervised. Supervised training involves a mechanism of providing the network with the desired output either by manually "distinguish" the network's performance or by providing the desired outputs with the inputs. Unsupervised training is where the network has to make sense of the inputs without outside help. Therefore more, we have three main classes of network architectures:

- – Single-layer Perceptron (SLP).
- – Multi-layer Perceptron (MLP).
- – Recurrent (Feedback).

5.1.1 Supervised training

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs, because in fact we will have two of the output of one of the actual and one is required (desired). Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked. The set of data which enables the training is called the "training set." During the training of a network the same set of data is processed many times as the connection weights are ever refined. The problems are resolved such as classification, recognition, diagnostic and regression. In addition to that the model such as perceptron, adaline, feed-forward and radial basis function.

5.1.2 Unsupervised training

The other type of training is called unsupervised training. In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption. The problems are resolved such as clustering and content addressable memory. In addition, the model such as Hopfield networks and self organizing maps.

5.1.3 Single-layer Perceptron (SLP)

This reflects structural one of the oldest structures in neural networks, which consists of one layer of neurons on the basis that the computational input layer does not undertake any operations of arithmetic. Associated with each neuron layer of input layer of all neuron in the output layer (fully connected), the figure (1.5) below shows the single- layer networks:

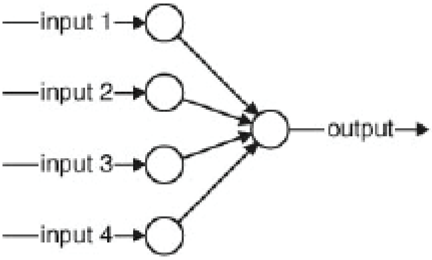


Fig. 1.5. A single-layer linear model

5.1.4 Multi-layer Perceptron (MLP)

There is input layer and the output layer, in addition to the many hidden layers. If the lines of communication between cells of the input layers moving towards the introduction hidden layers and then output layer then called these structures structure Feed-forward (feed-forward Architecture). This in addition to Single-layer Perceptron (SLP), the figure (1.6) below shows the multi – layer networks:

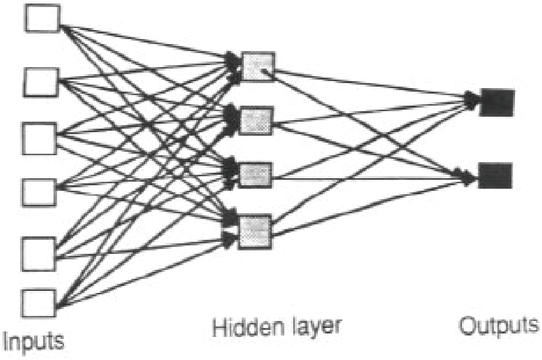


Fig. 1.6. A multi - layer liner model

5.1.5 Recurrent (Feedback)

Recurrent networks: can be unstable, or oscillate, or exhibit chaotic behavior e.g., given some input values, can take a long time to compute stable output and learning is made more difficult. However, can implement more complex agent designs and can model systems with state the Figure (1.7) below shows recurrent networks:

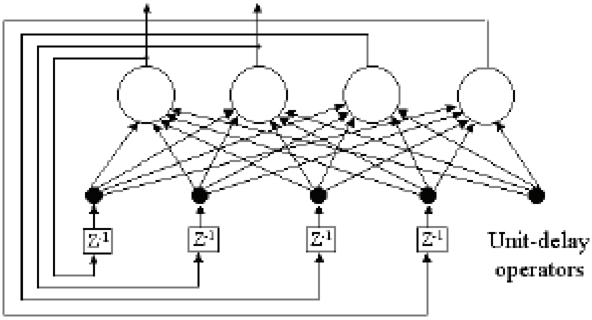


Fig. 1.7. A recurrent network with hidden neurons

5.2 Major components of an artificial neuron

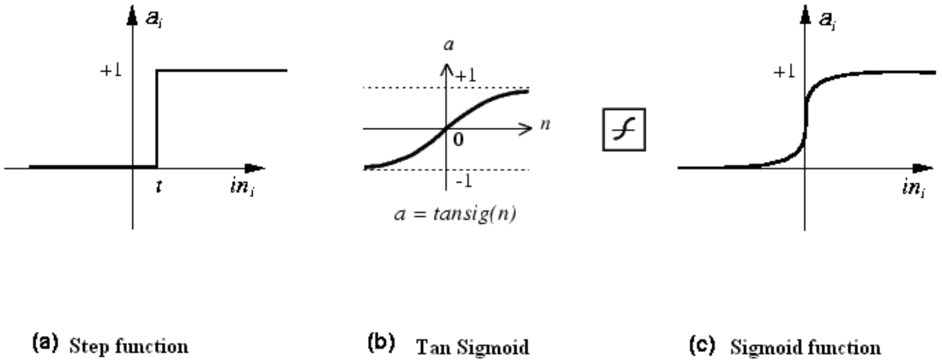
This section describes the seven major components which make up an artificial neuron. These components are valid whether the neuron is used for input, output, or is in one of the hidden layers:

a. Weighting Factors: A neuron usually receives many simultaneous inputs. Each input has its own relative weight which gives the input the impact that it needs on the processing element's summation function. These weights perform the same type of function as do the varying synaptic strengths of biological neurons. In both cases, some inputs are made more important than others so that they have a greater effect on the processing element as they combine to produce a neural response.

b. Summation Function: The first step in a processing element's operation is to compute the weighted sum of all of the inputs. Mathematically, the inputs and the corresponding weights are vectors which can be represented as (i1, i2 . . . in) and (w1, w2 . . . wn). The total input signal is the dot, or inner, product of these two vectors.

c. Transfer Function: The result of the summation function, almost always the weighted sum, is transformed to a working output through an algorithmic process known as the transfer function. In the transfer function the summation total can be compared with some threshold to determine the neural output. If the sum is greater than the threshold value, the processing element generates a signal. If the sum of the input and weight products is less than the threshold, no signal (or some inhibitory signal) is generated. Both types of responses are significant. In addition, the threshold, or transfer function is generally non-linear. Linear (straight-line) functions are limited because the output is simply proportional to the input. Linear functions are not very useful. The Figure (1.8) below shows the activation function

d. Scaling and Limiting: After the processing element's transfer function, the result can pass through additional processes which scale and limit. This scaling simply multiplies a scale factor times the transfer value, and then adds an offset.



(a) Step function (b) Tan Sigmoid (c) Sigmoid function

Fig. 1.8. sample transfer functions

e. Output Function (Competition): Each processing element is allowed one output signal which may output to hundreds of other neurons. This is just like the biological neuron, where there are many inputs and only one output action. Normally, the output is directly equivalent to the transfer function's result.

f. Error Function and Back-Propagated Value: In most learning networks the difference between the current output and the desired output is calculated. This raw error is then transformed by the error function to match particular network architecture.

g. Learning Function: The purpose of the learning function is to modify the variable connection weights on the inputs of each processing element according to some neural based algorithm. This process of changing the weights of the input connections to achieve some desired result can also be called the adaption function, as well as the learning mode.

5.3 Feed-forward Neural Network: Backpropagation (BP):

Most popular training method for neural networks, the generalized delta rule [Rum86], also known as Backpropagation algorithm which is explained here briefly for feed-forward Neural Network (NN). The explanation here is intended to give an outline of the process involved in Backpropagation algorithm. The (NN) explained here contains three layers. These are input, hidden, and output layer. During the training phase, the training data is fed into to the input layer. The data is propagated to the hidden layer and then to the output layer. This is called the forward pass of the Backpropagation algorithm. In forward pass, each node in hidden layer gets input from all the nodes from input layer, which are multiplied with appropriate weights and then summed. The output of the hidden node is the nonlinear transformation of this resulting sum. Similarly each node in output layer gets input from all the nodes of the hidden layer, which are multiplied with appropriate weights and then summed. The output of this node is the non-linear transformation of the resulting sum. The output values of the output layer are compared with the target output values. The target output values are used to teach network. The error between actual output values and target output values is calculated and propagated back toward hidden layer. This is called the backward pass of the Backpropagation algorithm. The error is used to update the connection strengths between nodes, i.e. weight matrices between input-hidden layers and hidden-output layers are updated. During the testing phase, no learning takes place i.e., weight matrices are not changed. Each test vector is fed into the input layer. The feed-forward of the testing data is similar to the feed-forward of the training data. Backpropagation architecture was developed in the early 1970s by several independent sources (Werbor; Parker; Rumelhart, Hinton and Williams). There are many laws (algorithms) used to implement the adaptive feedback required to adjust the weights during training. The most common technique is backward-error propagation, more commonly known as back propagation. The Backpropagation algorithm searches for weight values that minimize the total error of the network over the set of training examples (training set). Backpropagation consists of the repeated application of the following two passes:

- Forward pass: in this step the network is activated on one example and the error of (each neuron of) the output layer is computed.
- Backward pass: in this step the network error is used for updating the weights (credit assignment problem).

Therefore, starting at the output layer, the error is propagated backwards through the network, layer by layer. This is done by recursively computing the local gradient of each neuron. Here, a simple example shows the work of Backpropagation algorithm, uses supervised training, if the output is not correct, the weight are adjusted according to the formula:

$$W_{\mathbf{new}} = W_{\mathbf{old}} + \alpha \text{ (desired – output) * input.} \tag{3}$$

α is the learning rate, assume α = 1.
Assume output threshold = 1.2.
Figure (1.9) shows the neural network:

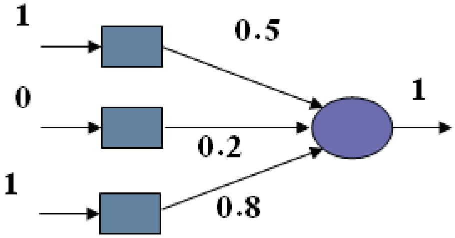


Fig. 1.9. sample neural network

To apply the above:

= 1 * 0.5 + 0.2 * 0 + 1 * 0.8
= 1.3
1.3 > 1.2

Then adjust the weight:

$W_{\mathbf{1new}}$ = 0.5 + 1 * (0 – 1) *1 = - 0.5
$W_{\mathbf{2new}}$ = 0.2 + 1 * (0 – 1) *0 = 0.2

$W_{3new} = 0.8 + 1 * (0 - 1) *1 = - 0.2$
$= 1 *(- 0.5) + 0.2 * 0 + 1 * (- 0.2)$
$= (- 0.5) + (- 0.2)$
$= - 0.7 \longrightarrow - 0.7 < 1.2$

Threshold, this is the edge and identify the value in the neural network.

6. Crisp set and Fuzzy set

Fuzzy logic (FL) was introduced by Dr. Lotfi Zadeh in the 1960 as a means to model the uncertainty of natural language [Zadeh 1965]. There are many motivation that encouraged scientists to develop the science of fuzzy logic or crisp logic, with the development of computer and software have the desire to reinvent or programmable systems can deal with inaccurate information along the lines of human, but this problem was born as the computer can not deal only with specific and accurate information, and has resulted in this trend known as expert systems or artificial intelligence. Knowledge of fuzzy logic is one of the theories through which they could build such systems. We can say in general that fuzzy logic as the fuzzy logic and fuzzy set. Fuzzy Logic (FL) is a multi valued logic, that allows middle values to be defined between conventional evaluations like true/false, yes/no, high/low, etc, this concept is sufficient for many areas of applications, but it can easily be seen, that it lacks in flexibility for some applications like classification of remotely sensed data analysis, with the fuzzy logic use the rule. In addition drawbacks of crisp set the membership function of crisp logic fails to distinguish between members of the same set, Figure (1.10) below shows the process:

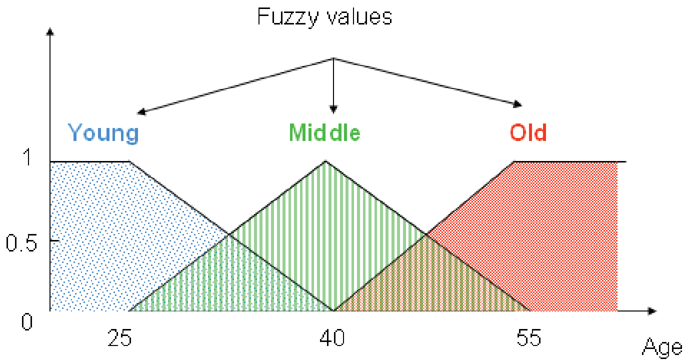


Fig. 1.10. the fuzzy logic process

We can note from Figure (1.10) the meaning of the Fuzzification scales and maps input variables to fuzzy sets, but the Defuzzification convert fuzzy output values to control signals. While in fuzzy sets an object can partially be in a set, the membership degree takes values between 0 and 1, 1 mean entirely in the set, 0 means entirely not in the set, other values mean partially in the set. The fuzzy sets are in the range [0.0, 1.0], with 0.0 representing absolute falseness and 1.0 representing absolute truth. This does not deal with rule but deal with the membership degree, we will deal with only fuzzy set, we will fuzzy the data only. Fuzzy set play an important role in recognizing dangerous events and reducing false alarms level [Yao and et al 2002]. Interest from the use the fuzzy set if the dataset are huge and complex (overlapping), the Fuzzification resolve this overlap. The fuzzy set deals with a linguistic variable associate's words or sentences with a measure of belief functions, also called membership function (use the natural language). The set of values that it can take is called term set, the figure (1.11) shows simple example of membership relation of age.

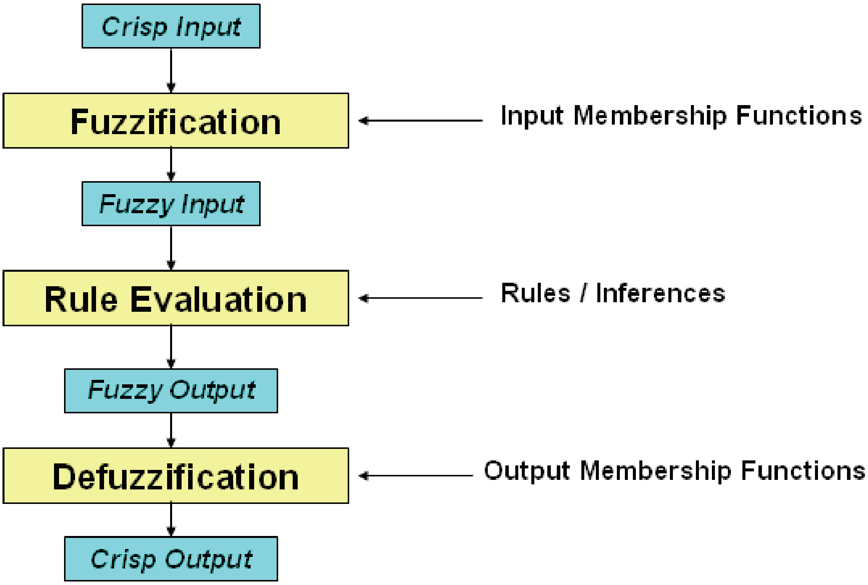


Fig. 1.11. Membership function

Through example, we can note the degree of affiliation of each and every one. Each value in the set is a fuzzy variable defined over a base variable.

7. Architecture of the Artificial Neural Networks

This part concerned with the explanation of the structural Artificial Neural Networks (ANN) used in this chapter, types and components have been clarified in the previous pages. In fact, in this work we used one type of neural network namely (feed-forward neural network), but with two different architectures, one trained with nonfuzzified dataset and other for the fuzzified data.

7.1 Artificial Neural Networks used with non-fuzzified data

In general we know that the feed-forward neural networks consists of three layers, these are input, hidden, and output layers. In this chapter, the feed-forward neural networks

(Backpropagation) are used. Input layer consists of twelve neurons (input) as equal features that have been selected from KDD dataset based on previous studies [Chebrolu and et al 2005]. The process of determining the number of hidden layers of each network is by using the concept (Trail and Error). Is to identify a certain number of layers and number of neurons per layer, therefore, different nets are trained and examining the network performance. Then choose the best network architecture. In this chapter, the hidden layer consists of 20 neurons. In addition to output layer consists of five neurons, this will be four of the intrusion (Probing, Dos, U2R, and R2L) and is one of the outputs is normal. In addition to the parameters used in this feed-forward Neural Network (NN) such as:

- TrainParam.epochs = 500; Epoch number (Batch, Sequential) – improve the result and condition stop.
- TrainParam.lr = 0.000001; learn rate, value to update weight – improve the result.
- TrainParam.goal = 0000; condition stop.
- TrainParam.min_grad=0.0000000001; value change in min_grad–improve the result.
- Threshold if the value is zero or less than zero is equal (-1), otherwise i.e. if the value more than zero is equal (1).

This is the type of neural network discussed briefly in previous pages: the units each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output, and the units are arranged in a layered feed-forward topology. An activation function used to transform the activation level of a unit (neuron) into an output signal. The Backpropagation algorithm uses an activation function. In this chapter, Tan-Sigmoid transfer function (tansig) is used, like the logistic function, except that output lies in the range (-1, +1). Often performs better than the logistic function because of its symmetry. Ideal for customization of multilayer perceptrons, particularly the hidden layers, Figure (1.12) below shows the Tan-Sigmoid function:

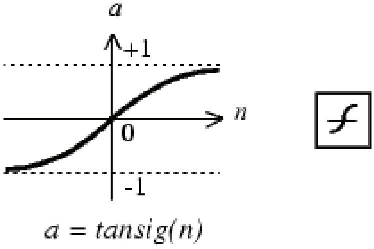


Fig. 1.12. Tan-sigmoid transfer function

Figure (1.13) shows the structure feed-forward neural network of that.

7.2 Artificial Neural Networks used with fuzzified data

In this type of feed-forward neural networks with fuzzified data, the input layer consists from sixty neurons, because of the membership, each value in the previous network will be represented with five values, the hidden layer consists of seven neurons. In addition to output layer consists of five neurons. In addition, the network will use the previous parameters and use the Backpropagation algorithm with the same transfer function. Figure (1.14) below shows the structure of the feed-forward neural network:

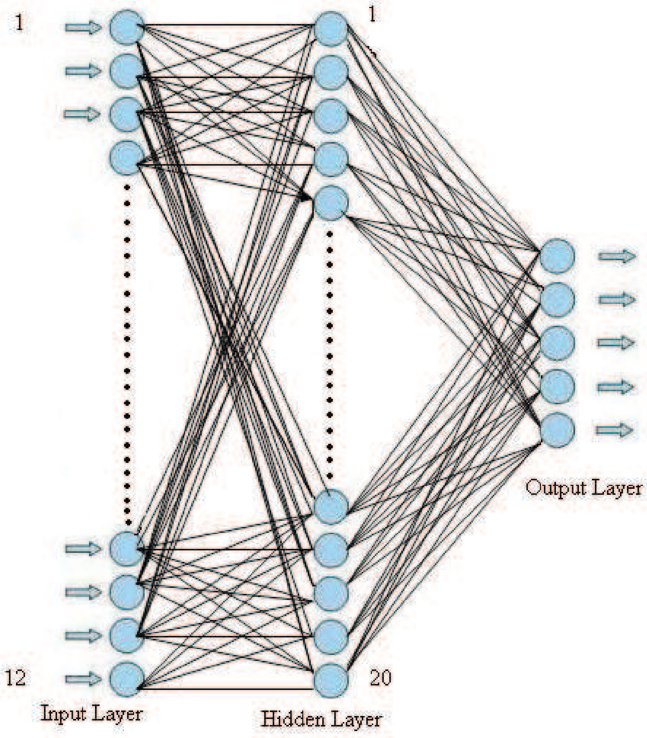


Fig. 1.13. Structure the neural network with normal data

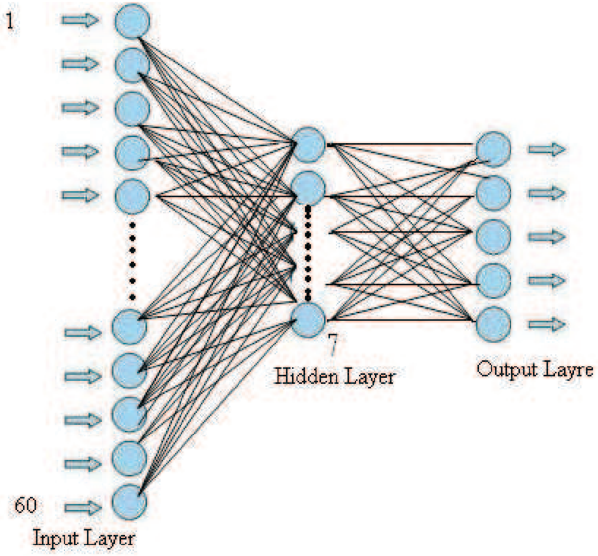


Fig. 1.14. Structure the neural network with Fuzzification data

8. Conclusions

The main contribution of the present chapter is to achieve a classification model with high intrusion detection accuracy and mainly with low false negative. This was done through the design of a classification process for the problem using neural network and neural network with Fuzzification data for the detection of various types of attacks.

From the given results, the following substantial remarks were obtained:

1. The neural networks were able to produce good detectors that give a good estimation of the amount of difference from the normal. This shows that it is possible to apply the suggested classifiers to detect anomalies on real network traffic data.
2. Two classifiers were suggested (Neural Network, Neural Network with Fuzzification data) with best classification accuracy 95.9303 % and 97.4890 % respectively. The suggested neural network with Fuzzification data is more efficient and meets the contribution of this chapter.
3. During the training process it was found that changing the size of the normal training data set with respect to the size of the attack data set (same percentages of the two data set or different percentages) did not affect the obtained classification.
4. Both anomaly detection and misuse detection are supported by the system. This was obvious from the ability of the designed NN (with fuzzified data) of giving approximately the same recognition ability (97.0038%) when tested with whole data set (311029) connection record) not only the trained set (30000 connection record) at which the testing result was (97.2700%). This shows that the designed net gives the ability to respond to anomalies and not only to signatures of known attacks.

The result of training and testing the neural network with Fuzzification data highly reduces the false negative alarms (to 0.804 %) compared with the NN trained with non fuzzified data set (the false negative rate was (1.9943 %) and that the false negative alarms only caused by R2L attack for fuzzified data set, while for non fuzzified data set caused by Prob, DOS, R2L.

9. Future work

1. Built an OCON (one class one neural) for each attack and study its ability to detected attacks and reduce the false alarms.
2. In this chapter, we used 12 features to classify connection records, as future work try to study the most suitable feature for each type of normal and attack, for example, in the normal use the feature number (3,11,21,40), in the probing use (1,3,12,18,20,21,23,26), in the Dos use (1,8,10,11,16,17,20,21,23,28,29,31), in the U2R use (11,14,17,28,29,32,36,38), and R2l use (1,3,11,12,18,20,22,25,38). This could be done with OCON structure
3. Another improvement of the specific intrusion system developed is the use of real-time intrusion detection, because such a system can catch a range of intrusion like viruses, Trojan horses, and masquerading before these attacks have time to do extensive damage to a system.
4. Design a classifier depending on the idea of data mining and compare its behavior with this chapter work.

10. References

[1] [Axe99] Stefan Axelsson, "Research in Intrusion –Detection System: A Survey ", Department of Computer Engineering, Chalmers University of Technology, Sweden. 1999, pp.1-25.

[2] [Bac01] Rebecca Bace & Peter Mell, "Intrusion Detection Systems", Released by National Institute of Standards and Technology (NIST), 2001 last accessed in 29-7-2007,pp,5-55. http://www.21cfrpart11.com/files/library/goverment/intrusion_detection_systems_0201 _draft PDF.

[3] [Che05] Chebrolu S, Abraham A, Thomas JP." Feature detection and ensemble design of intrusion detection systems". Compute Secur; 24:, 2005, pp.295–307.

[4] [Cro03] Tim Crothers, "Implementing Intrusion Detection Systems", Willey Publishing, 2003.

[5] [Den87] Dorothy E. Denning, "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, Vol.SE-13.No.2, February 1987, pp.222-232.

[6] [Kem02] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview", Computer, vol. 35, no. 4, 2002, pp. 27–30.

[7] [Kim02] Jung won Kim, "Integrating Artificial Immune Algorithms for Intrusion Detection", Dissertation in University of London, 2002, pp,1-5.

[8] [Kum95] Sandeep Kumar, "Classification and Detection of Computer Intrusions", A PHD Dissertation to the Faculty of Purdue University, August 1995, pp.49-51.

[9] [Mal02] By Saadat Malik," Network Security Principles and Practices", chapter 14, November 15, 2002.

[10] [Myk94] Mykerjee B., Heberlein L. T., & Levitt K. N., " Network Intrusion Detection", IEEE Network, Vol.8, No.3, 1994, pp. 26-14.

[11] [Por92] Porras, P. A., "STAT: A State Transition Analysis Tools for Intrusion Detection", MSc Thesis, Department of Computer Science, University of California Santa Barbara, 1992, pp.15-20.

[12] [Pri97] Katherine E. Price. "Host-based misuse detection and conventional operating system 'audit data collection ", A thesis Submitted to the Faculty of Purdue University, December 1997, pp. 6- 8.

[13] [Kaz04] Przemyslaw Kazienko & Piotr Dorosz,"Intrusion Detection Systems (IDS) part2- Classification; method; techniques", 2004, Web, last access day: 28-july-2007, http://www.windowsecurity.com/articles/IDS-part2-classification-methods-techniques.html.

[14] [Sto00] Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, & Philip K. Chan, "Cost-based modeling for Fraud and Intrusion Detection: Results from the JAM Project", 2000, http://www1.cs.columbia.edu/jam/recent-project-papers.html, last access day 28-july-2007, pp.1-15.

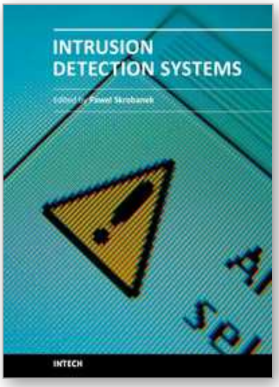
[15] [Sun02] Aurobindo Sundaram, "An Introduction to Detection ", ACM'c First Electronic Publication 2002.

[16] [Tao07] Tao Song, "Formal Reasoning about Intrusion Detection Systems", a dissertation submitted in partial satisfaction of the requirements for the degree of

doctor of philosophy in computer science in the office of graduate studies of the university of California, 2007.

[17] [Yao02] J.T Yao, S.L. Zhao, L.V. Saxton,”A study on Fuzzy Intrusion Detection“, Department of computer science, University of Regina. Saskatchewan, Canada S4S 0A2, 2002.

[18] [Zad65] L.A. Zadeh, Fuzzy Sets, “Information and Control”, 8(31965), pp.338-353, 1965.



Intrusion Detection Systems
Edited by Dr. Pawel Skrobanek

ISBN 978-953-307-167-1
Hard cover, 324 pages
Publisher InTech
Published online 22, March, 2011
Published in print edition March, 2011

The current structure of the chapters reflects the key aspects discussed in the papers but the papers themselves contain more additional interesting information: examples of a practical application and results obtained for existing networks as well as results of experiments confirming efficacy of a synergistic analysis of anomaly detection and signature detection, and application of interesting solutions, such as an analysis of the anomalies of user behaviors and many others.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Khattab M. Alheeti (2011). Intrusion Detection System and Artificial Intelligent, Intrusion Detection Systems, Dr. Pawel Skrobanek (Ed.), ISBN: 978-953-307-167-1, InTech, Available from: http://www.intechopen.com/books/intrusion-detection-systems/intrusion-detection-system-and-artificial-intelligent

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821