ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 4, April 2013

Soft Computing Based Classification Technique Using KDD 99 Data Set for Intrusion Detection System

Mr. Suresh kashyap[1] ,Ms. Pooja Agrawal[2], Mr.Vikas Chandra Pandey[3], Mr. Suraj Prasad Keshri[4]

Research Scholar (M.Tech.), Dr.C.V.RamanUniversity, Kargi Road Kota, Bilaspur,India[1]

Research Scholar (Ph.D.), Dr.C.V.RamanUniversity, Kargi Road Kota,Bilaspur,India[2]

Research Scholar (Ph.D.), Dr.C.V.RamanUniversity, Kargi Road Kota,Bilaspur,India[3]

Research Scholar (M.Tech.), Dr.C.V.RamanUniversity, Kargi Road Kota, Bilaspur,India[4]

Abstract*:* An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. Soft computing techniques resemble biological processes more closely than traditional techniques, which are largely based on formal logical systems. Knowledge Discovery in Databases (KDD) is the automated discovery of patterns and relationships in large databases. In this paper we are going to preprocess the different of KDD cup 99 data set. The two algorithms Error back propagation (EBP) which is the most used training algorithm for feedforwrd artificial neural networks (FFANNs) and the Radial basis function (RBF) neural network which is based on supervised learning are compared .After the process we give result that Radial basis function (RBF) is better than Error back propagation (EBP) .For comparison we used MATLAB tool.

Keywords: Detection methods, Matlab, intrusion detection, network security.

I.INTRODUCTION

An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. IDS' initial design and function is to protect the organization's vital information from an outsider. The IDS analyzes the information it gathers and compares it to large databases of attack signatures.

Intrusion detection functions include:-

- Monitoring and analyzing both user and system activities.
- Analyzing system configurations and vulnerabilities.
- Assessing system and file integrity.
- Ability to recognize patterns typical of attacks.
- Analysis of abnormal activity patterns.
- Tracking user policy violations.

II. IDS WITH TRADITIONAL APPROACH

The increasing complexity of modern computing systems makes traditional views of information security impractical, if not impossible. Computing environments are dynamic with near constant changes in configurations, software, and usage patterns. This makes completely securing a given system a difficult theoretical task for static Systems unfeasible for the dynamic nature of today's systems. This presents the need for a more dynamic view of information security, one that recognizes the insufficiency of static descriptions of policy and security mechanisms and that proposes a dynamic means of providing security which is sufficient for a given system at a given time.

Many draw back has in Traditional Approach:

- Signature-based IDSs must be programmed to detect each attack and thus must be constantly updated with signatures of new attacks.
- Many signature-based IDSs have narrowly defined signatures that prevent them from detecting variants of common attacks.
- Anomaly detection approaches usually produce a large number of false alarms due to the unpredictable nature of users and networks.

- Anomaly detection approaches often require extensive "training sets" of system event records in order to characterize normal behavior patterns.
- Application-based IDSs may be more vulnerable than host-based IDSs to being attacked and disabled since they run as an application on the host they are monitoring.

III.SOFT COMPUTING

Soft Computing became a formal Computer Science area of study in the early 1990. Earlier computational approaches could model and precisely analyze only relatively simple systems. More complex systems arising in biology, medicine, the humanities, management sciences, and similar fields often remained intractable to conventional mathematical and analytical methods

Components of soft computing include:-

- Neural networks (NN).
- Fuzzy systems (FS).
- Evolutionary computation (EC).
- Evolutionary algorithms.

A. Why Soft Computing Tools Used For IDS ?

Traditional protection techniques such as user authentication, data encryption, avoiding programming errors and firewalls are used as the first line of defense for computer security. If a password is weak and is compromised, user authentication can not prevent unauthorized use, firewalls are vulnerable to errors in configuration and suspect to ambiguous or undefined security policies. They are generally unable to protect against malicious mobile code, insider attacks and unsecured modems. Programming errors cannot be avoided as the complexity of the system and application software is evolving rapidly leaving behind some exploitable weaknesses. Consequently, computer systems are likely to remain unsecured for the foreseeable future. Intrusion detection is useful not only in detecting successful intrusions, but also in monitoring attempts to break security, which provides important information for timely countermeasures.

An Intrusion Detection System (IDS) itself can be defined as the tools, methods, and resources to help identify, assess, and report unauthorized or unapproved network activity.

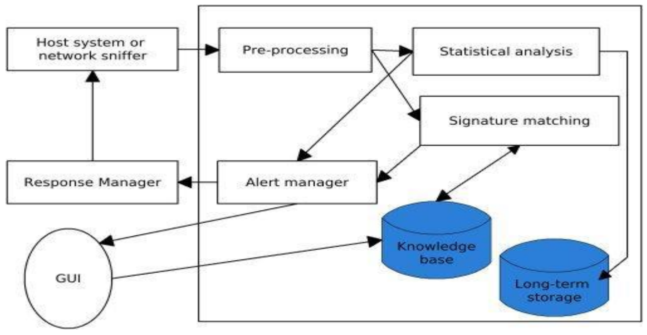

Figure 1 : Standard IDS system

IV.KDD 99 DATA SET

Knowledge Discovery in Databases (KDD) is the automated discovery of patterns and relationships in large databases. Large databases are not uncommon. Cheaper and larger computer storage capabilities have contributed to the proliferation of such databases in a wide range of fields.

KDD employs methods from various fields such as machine learning, artificial intelligence, pattern recognition, database management and design, statistics, expert systems, and data visualization. KDD has been more formally defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.The KDD Process is a highly iterative, user involved, multistep process, as can be seen in figure 2**.**

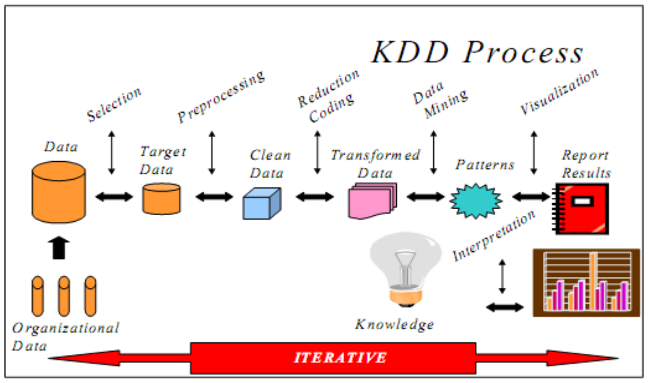

Figure 2: The KDD Process

We see that initially, we have organizational data. This data is the operational data gathered either in one or several locations.

V.CLASSIFICATION OF DATASET

The all dataset classifying the five broad on based of attacks categories these are:

- Normal dataset: The normal data set class means the IDS cannot detect any abnormal condition.
- DoS (Denial Of Service dataset): Denial of Service (DoS) is a class of attack where an attacker makes a computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine.
- R2L(Unauthorized Access from a Remote Machine dataset):A remote to user (R2L) attack 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. There are different types of R2U attacks; the most common attack in this class is done using social engineering.
- U2Su (Unauthorized Access to Local Super User (root) dataset): User to root exploits are 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. Most common exploits in this class of attacks are regular buffer overflows, which are caused by regular programming mistakes and environment assumptions.
- Probing (Surveillance and Other Probing dataset): Probing is a class of attack where an attacker scans a network to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use the information to look for exploits. There are different types of probes: some of them abuse the computer's legitimate features; some of them use social engineering techniques.

For training the KDD cup 99 data set we have given number to different types attack including normal attack as shown in table.

Attack type	**Class**	**Group**	**Sub attacks types**
Normal	1	A	normal
DoS	3	B	smurf, teardrop, pod, back, land, apache2,udpstrom,mailbomb, processtable, neptune
Probe	4	C	ipsweep, portsweep, nmap, satan, saint, mscan
R2L	2	D	dictionary,ftp_write,guess_password, imap, named, sendmail, spy, xlock, xsnoop, snmpgetattack, httptunnel, worm,snmpguess, multihop, phf, wraezclient, wrazemaster
U2R	5	E	perl, ps, xterm, loadmodule, eject, buffer_overflow, sqlattack

VI.CLASSIFICATION

Data classification is a methodology to align business requirements to infrastructure, so that infrastructure service delivery properly supports data storage and management.

Classification of objects is probably one of the most common and ancient decision tasks performed by humans. It can be seen as the ability of assigning a specific object to a predefined group or class based on a number of observed attributes of that object. The classification process was primarily related to our natural senses: humans recognize or classify objects based on the data acquired by their natural sensors. The data collected by the sensors is converted to specific features.

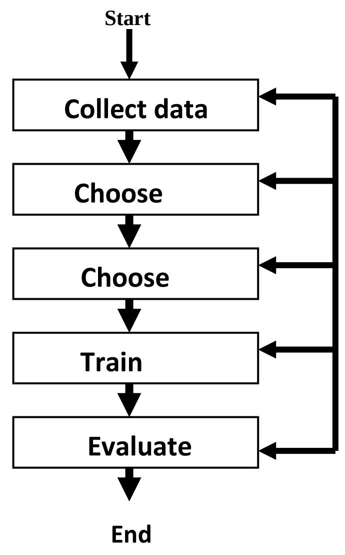

Figure 3 Classification Process

Above Figure 3 show the Schematic view of the Classification process.

VII.CLASSIFICATION THROUGH EBPA

One of the most popular weight updating rules of learning (training) algorithms is Error Back Propagation(EBP).However, most of the EBP based neural learning algorithms strictly depends on the architecture of the ANN and there are many problems associated with the currently existing algorithm based on EBP and its variation. Feed-Forward NN with EBP learning method are a very multi-purpose system. They can be seen as a statistical method, a non-linear controller, a filter, an agent behavior system and every other complex input-output function approximation and generalization.

Algorithm:

Training a neural net by back-propagation involves three stages:

• Feed-forward of input training pattern,

• Back-propagation of associated error, and

• Adjustment of weights.

The algorithm is as follows:

Step 1: Initialize the weights (set to random values).

Step 2: While stopping condition is false, do steps 2-9.

Step 3: For each training pair, do steps 3-8.

Feed forward:

Step 4: Each input unit (Xi, i = 1, 2,3, …., n) receives input signal xi and broadcasts this signal to all units in the layer above it i.e. the hidden layer.

Step 5: Each hidden unit (Zj, j=1,2,3……p) sums its weighted input signals,

$$z\_inj = v_{0j} + \sum_{i=1}^{n} x_i \, v_{ij},$$

applies its activation function to compute its output signal

$$z_j = f(z\_inj),$$

and sends this signal to all units in the layer above it i.e. the output layer.

Step 6: Each output unit (Yk, k=1,2,3……m) sums its weighted input signals,

$$y\_ink = w_{0k} + \sum_{j=1}^{p} z_j \, w_{jk},$$

applies its activation function to compute its output signal

$$y_k = f(y\_ink),$$

Back-propagation of error:

Step 7: Each output unit (Yk, k=1,2,3……m) receives a target pattern corresponding to the input training pattern, computes its error information term,

$$\delta_k = (t_k - y_k)\dot{f}(y\_ink),$$

calculates its weight-correction term (used to update w_{jk} later)

$$\Delta w_{jk} = \propto \delta_k z_j \ ,$$

calculates its weight-correction term (used to update w_{0k} later)

$$\Delta w_{0k} = \propto \delta_k,$$

and sends δ_k to units in the layer below.

Step 8: Each hidden unit (Zj, j=1,2,3……p) sums its delta inputs (from units in layer above),

$$\delta\_inj = \sum_{k=1}^{m} \delta_k \, w_{jk}$$

multiplies by the derivative of its activation function to calculate its error information term,

$$\delta_j = \delta\_inj * \dot{f}(z\_inj) \ ,$$

calculates its weight correction term (used to update vij later)

$$\Delta v_{ij} = \propto \delta_j x_i$$

and calculates its bias correction term (used to update v0j later)

$$\Delta v_{0j} = \propto \delta_j \, .$$

Update weights and biases:

Step 9: Each output unit (Yk, k=1 to m) updates its weight and bias (j = 0 to p):

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

Each hidden unit (Zj, j=1 to p) updates its weights and bias (i=0 to n)

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

Step 10: Test stopping condition.

VIII.CLASSIFICATION THROUGH RADIAL BASIS FUNCTION

Radial basis function (RBF) neural network is based on supervised learning. RBF networks were independently proposed by many researchers and are a popular alternative to the MLP. RBF networks are also good at modelling nonlinear data and can be trained in one stage rather than using an iterative process as in MLP and also learn the given application quickly. They are useful in solving problems where the input data are corrupted with additive noise.

Training of RBF neural networks:-

A training set is an m labelled pair $\{X_i, d_i\}$ that represents associations of a given mapping or samples of a continuous multivariate function. The sum of squared error criterion function can be considered as an error function E to be minimized over the given training set. That is, to develop a training method that minimizes E by adaptively updating the free parameters of the RBF network. These parameters are the receptive field centres μ_j of the hidden layer Gaussian units, the receptive field widths σ_j, and the out-put layer weights (w_{ij}). Because of the differentiable nature of the RBF network transfer characteristics, one of the training methods considered here was a fully supervised gradient-descent method over E.

In particular, μ_j $\sigma_j$and w_{ij} are updated as follows:

$$\Delta\mu_j = -\rho_\mu \nabla_{\mu_j} E, \qquad (4)$$

$$\Delta\sigma_j = -\rho_\sigma \frac{\partial E}{\partial \sigma_j}, \qquad (5)$$

$$\Delta w_{ij} = -\rho_w \frac{\partial E}{\partial w_{ij}}, \qquad (6)$$

where ρ_μ, ρ_σ and ρ_W, are small positive constants. This method is capable of matching or exceeding the performance of neural networks with back-propagation algorithm, butgives training comparable with those of sigmoidal type of FFNN14.

The training of the RBF network is radically different from the classical training of standard FFNNs. In this case, there is no changing of weights with the use of the gradient method aimed at function minimization. In RBF networks with the chosen type of radial basis function, training resolves itself into selecting the centres and dimensions of the functions and calculating the weights of the output neuron. Now simulate IDS data through MATLAB s/w using EBPA and RBN then we getting the following result

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 4, April 2013

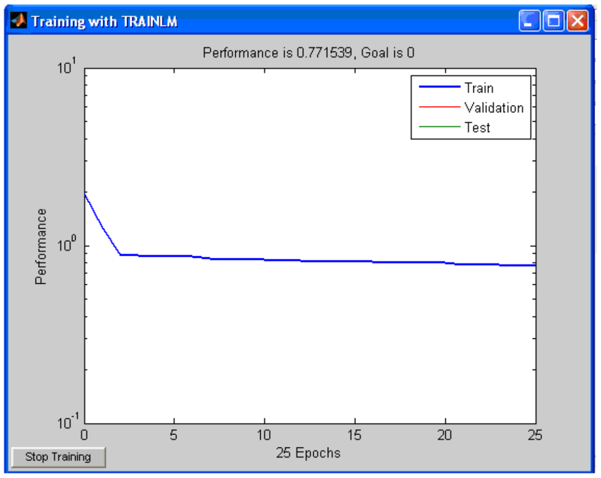

Figure 4: Training with TRAINLM window for input data

Target data	[1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,3,3,3,3,3,3,3,2,2,2,2,2,2,2,2,2,2,4,4,4,4,4,4,4,4,4,4];
Output data from EBP	[1 1 1 1 1 1.0007 2.4723 2.4723 2.4723 1.3301 2.4723 2.976 2.4722 2.7788 2.4723 2.4723 2.4723 2.4723 2.7788 2.4723 2.4723 1.5577 2.0004 2.0005 1.993 1.9961 1.9808 1.9879 1.9941 1.9808 1.986 4 3.9988 4 3.9882 4 4 4 3.9747 3.9692 3.9667]
Output data from RBF	[1 1 1 1 1 1 2.4 2.4 2.4 1 2.4 3 3 3 2.4 2.4 2.4 2.4 3 2.4 2.4 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4]

IX.COMPARISON

As we have trained our Neural network using EBP Algorithm and RBF and we are getting different output .It is clear from above two figurer but for IDS data (training) RBF is working well while EBP has shown less efficient result.

We have tried and trained our Neural network for very less amount of data , this may be the reason why me are getting error full result . for getting error less result we can perform following task .

1) Initializing better weight of connection of Neural Network .
2) Setting another parameter like bias neuron .
3) Considering more number of training data of IDS .

X.CONCLUSION AND FURTHER RESEARCH

This paper consist the training of Neural Network specially designed for IDS data lots of works has been done in this field number of soft computing based tools were designed for Intrution detection, This paper is an effort towards simulation of IDS data for developing intelligent system for IDS. Our result show that this approach of developing IDS can be enhanced by using different technique an discussed in comparison parts.

Further research in this field can be carried out by considering different algorithm for training neural network with more amount of data and to compare and conclude me result that which algorithm will be suitable for IDS data, further a new soft computing tool can be designed for IDS system using hybrid technology of soft computing .

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 4, April 2013

REFERENCES

1. William Stallings " Cryptography and network security" Pearson prentice hall 2009 Fourth edition
2. An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition
3. Neural Networks at Pacific Northwest National Laboratory
4. http://www.emsl.pnl.gov:2080/docs/cie/neural/neural.homepage.html
5. Ajith Abraham1 and Ravi. Jain2 "Soft Computing Models for Network Intrusion Detection Systems" 1 Department of Computer Science, Oklahoma State University, USA ajith.abraham@ieee.org 2 School of Information Science, University of South Australia, Australia ravi.jain@unisa.edu.au
6. An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition
7. Simon Haykin "Neural network" Pearson prentice hall 2008 Second edition
8. Dubois, D. and Prade, H., 1980. Fuzzy Sets and Systems: Theory and Applications. New York: Academic
9. Kosko, B., 1991. Neural Networks and Fuzzy Systems. Englewood Cliffs, NJ: Prentice Hall.
10. IC1043 "Neural network & fuzzy logic" RMKEC page no 1to39
11. kddcup.names A list of features.
12. http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data.gz The full data set (18M; 743M Uncompressed)
13. http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz A 10% subset. (2.1M; 75M Uncompressed)
14. http://kdd.ics.uci.edu/databases/kddcup99/kddcup.newtestdata_10_percent_unlabeled.gz (1.4M; 45M Uncompressed)
15. http://kdd.ics.uci.edu/databases/kddcup99/kddcup.testdata.unlabeled.gz (11.2M; 430M Uncompressed)
16. http://kdd.ics.uci.edu/databases/kddcup99/kddcup.testdata.unlabeled_10_percent.gz (1.4M;45M Uncompressed)
17. http://kdd.ics.uci.edu/databases/kddcup99/corrected.gz Test data with corrected labels.
18. training_attack_types A list of intrusion types.
19. typo-correction.txt A brief note on a typo in the data set that has been corrected.
20. http://www.sigkdd.org/kddcup/index.php?section=1999&method=data
21. http://matauranga.wordpress.com/rana/kdd-cup-1999-data-evaluation/
22. http://www.scribd.com/doc/2346440/Neural-Networks-and-Fuzzy-logic-Control-IC-1403