Intrusion Detection and Attack Classification using Back-propagation Neural Network

Roshani Gaidhane
M.Tech Student of CSE Department
RGCER, RTMNU University
Nagpur, India

C. Vaidya
Lecturer in CSE Department
RGCER, RTMNU University
Nagpur, India

Dr. M. Raghuwanshi
Principal
RGCER, RTMNU University
Nagpur, India

Abstract— **Intrusion detection is a process that analyzes abnormalities in system or network activities. For security purpose it is necessary to identify malicious events correctly. Majority of research is going on neural network and machine learning technique for detecting intrusions. Several researchers used back propagation neural network approach for their experimentation. Mostly they have used KDDCup'99 dataset and classified the events into major attack classes i.e. DoS, U2R, R2L and Probe. But for security experts it is necessary to identify the attack type to quickly take particular action on it. Therefore the current research work is to detect and classify instance into its specific attack type. In this research paper, using KDDCup'99 dataset, instances are classified into 23 attack types. Back propagation neural network (BPN) classifier is built for classification with the help of "Waikato Environment for Knowledge Analysis (WEKA)" library and evaluated by observing detection rate. Results showed that it classifies instances into several attack types with low detection rate.**

Keywords— anomaly detection; attack classification; back propagation; classifier; detection rate;; intrusion detection system (IDS); neural network

I. INTRODUCTION

Intrusion detection system (IDS) [1] watches for abnormalities in traffic and raises alarm. There are two kinds of IDS i.e. Host based IDS and Network based IDS [2], [3]. The basic techniques used for intrusion detection are anomaly detection and misuse or signature based detection [4], [5], [6]. Anomaly detection watches abnormalities in traffic whereas misuse detection tries to match data with known attack pattern. One of the major disadvantage of misuse detection [5] is new form of attack is not detected. Therefore most researches focused on anomaly detection techniques [3]. Anomaly based technique has statistical, neural network, machine learning [7] and data mining [8], immune system approaches[6], [9].

Neural network approach has the ability to detect known and unknown attacks. It can be distinguished into supervised and unsupervised training algorithm. Multilayer perceptron (MLP) is the model for supervised algorithm whereas the Self organisation map (SOM) is unsupervised algorithm. Several research papers have used neural network approach [10]. According to survey in [11], back propagation neural network (BPN) has good detection rate as compared to other neural network techniques and therefore it can be used for specific attack classification, so that preventive action can be taken. BPN uses supervised learning approach for training. According to experimental results of previous research papers, it has been seen that BPN achieved good results

Aim of this paper is to build predictive model using back propagation algorithm to detect and classify attack type. First BPN network is trained using KDD'Cup99 dataset with 23 attack types. After training it will do predictions on test data to classify the events into its attack types.

The paper is organized in five sections. Section II introduces previous research work. Section III of the paper describes methodology and the results of the experimentation given in section IV. Finally paper is concluded with Section V.

II. PREVIOUS WORK

Previous results of neural network approach are described in this section.

V. Jaiganesh, Dr. P. Sumathi, S. Mangayarkarasi [12], have classified attacks into four classes DoS, Probe, U2R, R2L using Machine learning and BPN techniques. They have worked on detection rates for four attacks. For DoS attack the detection rate is 78.15% using BPN.

Changjun Han, Yi Lv and Dan Yang, Yu Hao in [13], trained data using BPN model with 8 attack types. Where, 1325 connections used for training and 1245 for testing. Their obtained results are: detection rate 80.5%, false alarm rate 7.4% and omission rate 11.3%. Sufyan T. Faraj and et al. in [14] , first trained data to detect and classify normal and abnormal events using BPN. Then abnormal events are further classified into five categories. Detection rate and false positive rate is calculated in different scenarios. For detection of normal and abnormal events detection rate for test set is about 90% and for classification into DoS, U2R, R2L, Probe is approximately 60-85%.

I Mukhopadhyay and etl al [15], trained BPN Neural Network Model for DoS, U2R, Probe, U2L and normal attack classes. The system gets success rate 73.9% for new test set and 95.6 % for level 1 test set. Hua TANG and Zhuolin CAO in [16] used SVM and MLP neural network for anomaly detection. They compared accuracy for DoS, U2R, Probe, U2L attack classes and found that accuracy of neural network is better than SVM. Vladimir Bukhtoyarovf and Eugene Semenkin [17], used neural network ensemble approach. Their work was focused on classifying probe attacks using joint usage of trained neural network. They found 99.87% detection rate for probe attacks but large amount of training time required which was one of the IDS issue.

It has been seen that Most of the researchers classified events into major attack classes and got good results for back propagation [12],[13],[14],[15],[18]. It is found that BPN is efficient to build IDS in [15]. In this paper, the research work is to detect, classify events into its specific attack type using back propagation and evaluate the system by observing detection rate.

III. METHODOLOGY

First collected data from KDDCup'99 intrusion detection and evaluation dataset [19]. Selected small portion of training and testing data from KDDCup'99 Dataset for experiment. Then pre-processing [20] is carried out. BPN classifier is built for detection and classification of events. Data trained using BPN [13] training algorithm. After training and testing, it classifies the connections into 23 categories (22-attack types and normal)..

A. *KDD CUP'99 Dataset*

It is a subset of DARPA 1998 Intrusion Detection and Evaluation Dataset. The dataset is used for the evaluation of computer network intrusion detection system. It consists of normal and attack records. Each record consists of 41 features and 1 class attribute. Class attribute specifies the nature of record (i.e. either it is attack or normal record). Used 10%KDD for training. It contains 22 attack types grouped into four attack classes i.e. DoS, U2R, R2L and Probe [21],[15] shown in TABLE I.

TABLE I. ATTACK CLASSES AND ITS TYPES

Attack Class	Attack Type
DoS	back, smurf, neptune, teardrop, pod, land
Probe	satan, ipsweep, portsweep, nmap
R2L	warezclient, warezmaster,guess_passwd, imap, ftp_write, multihop, phf, spy
U2R	buffer_overflow, rootkit,loadmodule,perl

In this paper, network instances are going to classify into 23 categories (22 attack and normal records)

TABLE II and TABLE III shows training and testing records, which are used for experimentation. For training, total 1279 connections are used from 10%KDD and 1183 connections for testing. Testing data are chosen from corrected test KDD and whole KDD dataset.

TABLE II. TRAINING SET

Attack Type	No. of Records
back	100
Buffer_overflow	30
ftp_write	8
guess_passwd	53
imap	12
ipsweep	100
land	21
loadmodule	9
multihop	7
neptune	100
nmap	100
normal	100
perl	3
phf	4
pod	100
portsweep	100
rootkit	10
satan	100
smurf	100
spy	2
teardrop	100
warezclient	100
warezmaster	20

TABLE III. TESTING SET

Attack Type	No. of Records
back	100
Buffer_overflow	22
ftp_write	3
guess_passwd	100
imap	1
ipsweep	100
land	9
loadmodule	2
multihop	18
neptune	100
nmap	84
normal	100
perl	2
phf	2
pod	87
portsweep	100
rootkit	13
satan	100
smurf	100
spy	2
teardrop	12
warezclient	26
warezmaster	24

B. *Preprocessing*

First collected following string features from the data.

Protocol_type=icmp,udp,tcp

Attack=phf,buffer_overflow,teardrop,guess_passwd,multihop, loadmodule,smurf,spy,normal,land,back,portsweep,warezclient,ftp_write,nmap,satan,rootkit,perl,imap,neptune,warezmaster, ipsweep,pod

Flag=RSTR,S3,SF,RSTO,SH,OTH,S2,RSTOS0,S1,S0,REJ

Service=vmnet,smtp,ntp_u,shell,kshell,aol,imap4,urh_i,netbios_ssn,tftp_u,mtp,uucp,nnsp,echo,tim_i,ssh,iso_tsap,time,netbios_ns,systat,hostnames,login,efs,supdup,http_8001,courier, ctf,finger,nntp,ftp_data,red_i,ldap,http,ftp,pm_dump,exec,klogin,auth,netbios_dgm,other,link,X11,discard,private,remote_job,IRC,daytime,pop_3,pop_2,gopher,sunrpc,name,rje,domain ,uucp_path,http_2784,Z39_50,domain_u,csnet_ns,whois,eco_i,bgp,sql_net,printer,telnet,ecr_i,urp_i,netstat,http_443,harvest

Then transformed data to remove string and replaced it by a 0, 1 numeric representation. For ex. If TCP occurs then it will converted into 001, if S3 flag occurs then it will converted into 01000000000 and so on. Then scaled the transformed file and converted all data within 0 to 1 range. After preprocessing, each row of 41 features and 1 class attribute is converted into 122, 23 numeric pattern respectively.

C. *Back-propagation Classifier*

Used BPN algorithm [13] to build BPN classifier for classification of events.

- Back Propagation Neural Network Algorithm (BPN):

  Step1) Design Network and set parameters

  Step2)Initialize weights with random values.

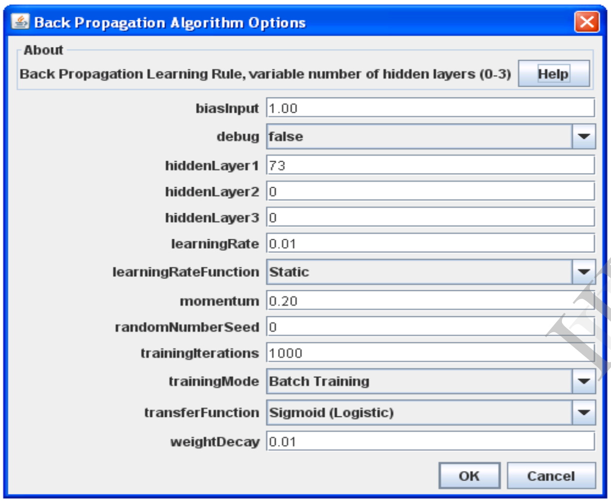
  For a specified number of training iterations do:

  For each input and ideal (expected) output pattern

  - i) Calculate the actual output from the input
  - ii) Calculate output neurons error
  - Calculate hidden neurons error
  - iii) Calculate weights variations (New wt)

  Step3) Learn by new weights.

For BPN, parameters are set as shown in Fig. 1.



Back Propagation Learning Rule, variable number of hidden layers (0-3)

biasInput 1.00
debug false
hiddenLayer1 73
hiddenLayer2 0
hiddenLayer3 0
learningRate 0.01
learningRateFunction Static
momentum 0.20
randomNumberSeed 0
trainingIterations 1000
trainingMode Batch Training
transferFunction Sigmoid (Logistic)
weightDecay 0.01

Fig. 1. BPN Parameters

Where no. of input and output neurons are 122 and 23 respectively. Fig. 2 shows BPN neural network model used for experimentation. Each output neuron is for particular attack type.

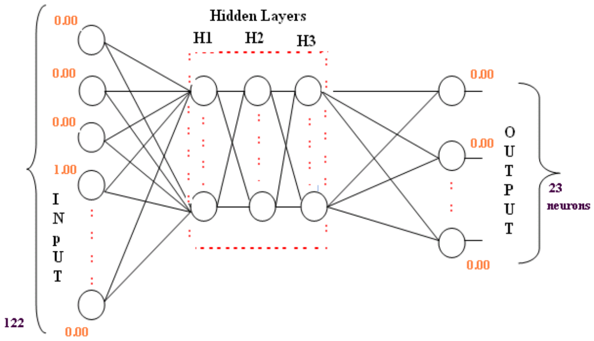


Fig. 2. BPN architecture

WEKA [22] library is used for building BPN classifier. As it takes training file in ARFF format, we have converted scaled file into ARFF format.

To measure the performance of the classifier, detection rate is calculated by following formula.

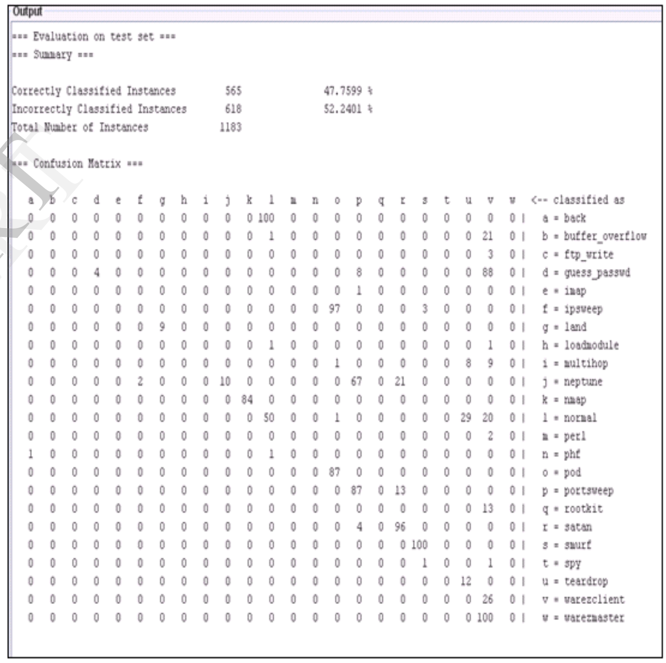
$$\text{Detection rate (Dr)} = (m_d \div n) \times 100 \ [13]$$

Where, m_d is no. of correctly classified instances and n represents total no. of instances

IV. EXPERIMENTAL RESULTS

Trained data with 22 attack types and normal connections. Then tested with test set shown in TABLE III. Number of experimentation is carried out for training the network with best parameters. BPN network consists of 122 input nodes, 73 hidden neurons and 23 output neurons. It takes 1000 epochs to train with learning rate 0.01. Total time required to build the model is 239.27 seconds.

Fig.3 depicts the classification output. Confusion matrix [10] used in Fig.3 shows actual vs. predicted attack type.



```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances         565               47.7599 %
Incorrectly Classified Instances       618               52.2401 %
Total Number of Instances             1183

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q   r   s   t   u   v   w   <-- classified as
   0   0   0   0   0   0   0   0   0   0   0 100   0   0   0   0   0   0   0   0   0   0   0 |   a = back
   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0  21   0 |   b = buffer_overflow
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0 |   c = ftp_write
   0   0   0   4   0   0   0   0   0   0   0   0   0   0   0   8   0   0   0   0   0  88   0 |   d = guess_passwd
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0 |   e = imap
   0   0   0   0   0   0   0   0   0   0   0   0   0   0  97   0   0   0   3   0   0   0   0 |   f = ipsweep
   0   0   0   0   0   0   9   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   g = land
   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0 |   h = loadmodule
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   8   9   0 |   i = multihop
   0   0   0   0   0   2   0   0   0  10   0   0   0   0   0  67   0  21   0   0   0   0   0 |   j = neptune
   0   0   0   0   0   0   0   0   0   0  84   0   0   0   0   0   0   0   0   0   0   0   0 |   k = nmap
   0   0   0   0   0   0   0   0   0   0   0  50   0   0   1   0   0   0   0   0  29  20   0 |   l = normal
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0 |   m = perl
   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0 |   n = phf
   0   0   0   0   0   0   0   0   0   0   0   0   0   0  87   0   0   0   0   0   0   0   0 |   o = pod
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  87   0  13   0   0   0   0   0 |   p = portsweep
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  13   0 |   q = rootkit
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0  96   0   0   0   0   0 |   r = satan
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100   0   0   0   0 |   s = smurf
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0 |   t = spy
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  12   0   0 |   u = teardrop
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  26   0 |   v = warezclient
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 100   0 |   w = warezmaster
```

Fig.3. Classification output

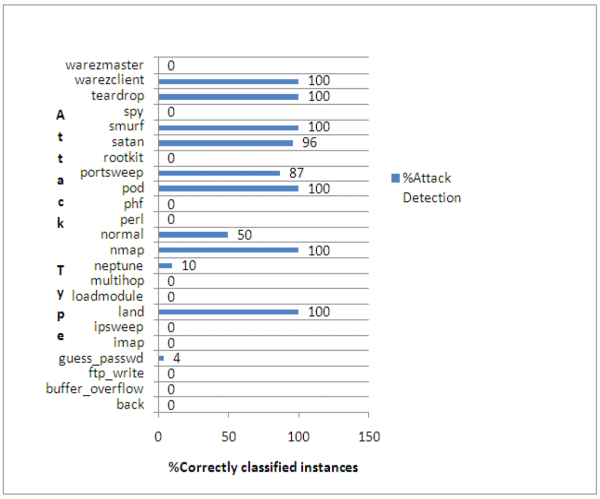


Fig.4. Attack detection chart

Chart in Fig. 4 shows %detection rate for each attack type. Out of total 1183 instances 565 instances are correctly classified. Overall detection rate is 47.75%.

From the results it has been observed that BPN has low detection rate if the events are classified into specific attack type.

V. CONCLUSION AND FUTURE WORK

In this paper, intrusion detection and classification is done using back propagation neural network approach. The instances are classified into 22 attack types and normal categories. From the results it has been observed that, for specific attack type classification BPN gives low detection rate. There is more to do to solve detection rate issue. To improve the results extreme learning machine (ELM) approach can be used in future.

ACKNOWLEDGMENT

All Authors would like to thank to the people who introduces KDDCup'99 intrusion detection Dataset. Implementation in this research paper is made possible with the help of WEKA machine learning library.

REFERENCES

[1] Rozenblum D., "Understanding Intrusion Detection Systems",SANS Institute Reading Room site1

[2] Rajasekhar K., Sekhar Babu B. ,Lakshmi Prasanna P., Lavanya D.R.,Vamsi Krishna T. "An Overview of Intrusion Detection System Strategies and Issues",International Journal of Computer Sci ence & Technology Vol. 2, Iss ue 4, Oct . - Dec. 2011.

[3] Sonawane S., Pardeshi S. and Prasad G.,"A survey on intrusion detection techniques",World Journal of Science and Technology 2012, 2(3):127-133

[4] Vinod Kumar,D. Sangwan P., "Signature Based Intrusion Detection System Using SNORT", International Journal of Computer Applications & Information Technology, Vol. I, Issue III, November 2012 (ISSN: 2278-7720)

[5] Uddin M., Khowaja K. and Rehman A., "Dynamic Multi-Layer Signature Based Intrusion Detection System Using Mobile Agents", International Journal of Network Security & Its
Applications (IJNSA), Vol.2, No.4, October 2010

[6] Ning P., Jajodia S.,"Intrusion Detection Techniques", http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.2492&rep=rep1&type=pdf

[7] Vinchurkar D. and Reshamwala A., "A Review of Intrusion Detection System Using Neural Network and Machine Learning Technique",International Journal of Engineering Science and Innovative Technology (IJESIT),Volume 1, Issue 2, November 2012

[8] Lee W., Stolfo S.J., and Mok K., “Adaptive Intrusion Detection: A Data Mining Approach,” Artificial Intelligence Rev., vol. 14, no. 6,pp. 533-567, Kluwer Academic Publishers, Dec. 2000

[9] Kim J., Bentley P.,Aickelin U., Greensmith J., Tedesco G., Twycross J.,"Immune System Approaches to Intrusion Detection - A Review",Natural Computing, Springer, in print, doi: 10.1007/s11047-006-9026-4, pp TBA.

[10] Xiaonan Wu S. and Banzhaf W.," The Use of Computational Intelligence in Intrusion Detection Systems: A Review", A technical report #2008-05,Memorial University of Newfoundland, St John’s, NL A1B 3X5, CA

[11] Shah B. and Trivedi H. ,"Artificial Neural Network based Intrusion Detection System: A Survey",International Journal of Computer Applications (0975 – 8887),Volume 39– No.6, February 2012

[12] Jaiganesh V., Sumathi P. and Mangayarkarasi S.,"An Analysis of Intrusion Detection System using Back Propagation NeuralNetwork",IEEE 2013 publication

[13] Han C., Yi Lv, Yang D., Hao Y., "An Intrusion Detection System Based on Neural Network",2011 International Conference on Mechatronic Science, Electric Engineering and Computer, August 19-22, 2011, Jilin, China,IEE Publication

[14] Faraj S, Al-Janabi and Saeed H, "A Neural Network Based Anomaly Intrusion Detection System",2011 Developments in E-systems Engineering,DOI 10.1109/DeSE.2011.19,IEEE publication

[15] Mukhopadhyay I, Chakraborty M, Chakrabarti S, Chatterjee T, "Back Propagation Neural Network Approach to Intrusion Detection System",2011 International Conference on Recent Trends in Information Systems, IEEE publication

[16] Tang H. and Cao Z., "Machine Learning-based Intrusion Detection Algorithms", Journal of Computational Information Systems5:6(2009) 1825-1831

[17] Bukhtoyarovf V. and Semenkin E., “Neural Networks Ensemble Approach for Detecting Attacks in Computer Networks,” WCCI 2012 IEEE World Congress on Computational Intelligence,June, 10-15, 2012 - Brisbane, Australia

[18] Chang R., Lai L.,Su W., Wang J., Kouh J., "Intrusion Detection by Backpropagation Neural with Sample-Query and Attribute-Query",International Journal of Computational Intelligence Research, ISSN 0973-1873 Vol.3, No. 1 (2007), pp. 6-10

[19] KDD Cup 1999 Data, https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[20] Hsu C.,Chang C., and Lin C., "A Practical Guide to Support Vector Classification", http://www.csie.ntu.edu.tw/~cjlin

[21] Eldos T., Siddiqui M. and Kanan A., "On The KDD'99 Dataset: Statistical Analysis for Feature Selection", Journal of Data Mining and Knowledge Discovery, Volume 3, Issue 3, 2012, pp.-88-90**.**

[22] Weka-TheUniversityofWaikato, http://weka.wikispaces.com/Use+WEKA+in+your+Java+code