Novel Intrusion Detection System integrating Layered Framework with Neural Network

Nidhi Srivastav
Dept. of Computer Science &Engg.
NITTTR Chandigarh
Chandigarh, India
nidhi3srivastav@yahoo.co.in

Rama Krishna Challa
Dept. of Computer Science &Engg.
NITTTR Chandigarh
Chandigarh, India
rkc_97@yahoo.com

Abstract—**The threat from spammers, attackers and criminal enterprises has grown with the expansion of Internet, thus, intrusion detection systems (IDS)have become a core component of computer network due to prevalence of such threats. In this paper, we present layered framework integrated with neural network to build an effective intrusion detection system.This system has experimented with Knowledge Discovery & Data Mining(KDD) 1999 dataset. The systems are compared with existing approaches of intrusion detection which either uses neural network or based on layered framework. The results show that the proposed system has high attack detection accuracy and less false alarm rate.**

Keywords—IDS; neural network; layered framework;KDD cup99 dataset

I. INTRODUCTION

Intrusion detection system(IDS) is a tool that is being used to protect organization from attacks from different sources. Intrusion detection systemhave emerged in the computer security area because of the difficulty of ensuring that an information system will be free of security flaws. Intrusion detection is defined by the Sysadmin, Audit, Networking and Security (SANS) institute as the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource[1]. Thus, security of data and continuity of services can only be ensured by IDS. It is required that IDS can handle large amount of data without affecting performance and without dropping data and can detect attacks reliably without giving false alarms.

An IDS is broadly classified as:

A. *Misuse based system*[1]

In misuse based IDS, detection is performed by looking for the exploitation of known weak points in the system, which can be described by a specific pattern or sequence of events or data. That means these systems can detect only known attacks for which they have a defined signature.

B. *Anomaly based system*[1]

In anomaly based IDS, detection is performed by detecting changes in the patterns of utilization or behavior of the system. The main advantage of anomaly detection system is that they can detect previously unknown attacks.

After the introduction in Section I, related work and its associated problems are described in Section II. Section III describes architecture of the proposed systems. Section IV explains the dataset, attack types & features used for classifying connection records. Section V shows the details of the experimental setup and results. Section VI concludes the paper with a discussion of results and scope of future work.

II. RELATED WORK

Neural network algorithms are emergingnowadays as a new artificial intelligence technique that canbe applied to real-life problems. Neural networks are a form of artificial intelligence that uses multiple artificial neurons, networked together to process information. This type of network has the capability to learn from patterns, and extrapolate results from data that has been previously entered into the network's knowledge base. This ability makes neural network applications extremely valuable in intrusion detection.[2]presents an approachof user behavior modeling that takes advantage of theproperties of neural network algorithms coupled with expert system.[3] usesmultilayer hierarchical KohonenNet, or Kohonen self-organizing map (K-Map) to implementan anomaly based intrusion detection system. In [4], [5] neural networks havebeen applied to build keyword-count-based misuse detectionsystems. The data presented to the systems consist of attackspecifickeyword-counts in network traffic. Authors in [6] also uses neural network to analyze program behavior profiles forboth anomaly detection and misuse detection to identify thenormal system behavior. [7] presents the results of a study on intrusion detection on IIS(Internet information services) utilizing a hybrid intrusion detection system. Thefeasibility of the hybrid IDS is validated based on the Internet scanner system (ISS) in 2005. [8] applies neural network approach to probingattacks that are the basis of other attacks in computer network systems in 2009.Authors in [9] showneural networkpattern recognition back propagation algorithm to be effective in intrusion detection.Comparing different neural networkclassifiers, the back-propagation neural network (BPN) isshown to be more efficient in developing IDS. However, thesimulation time required to induce models from large datasets islong.

978-1-4673-4529-3/12/$31.00 ©2012 IEEE

IDS also uses different kind of frameworks,[10] uses stacking framework which is constructed by ensembling of heterogeneous classifiers. The authors show that the output from these classifiers can be combined to generate a better classifier rather than selecting the individual best classifier. Similarly in [11],combination of 'weak' classifiers are used where the individual classification power of weak classifiers is shown to be slightly better than that of random guessing. Authors in [12] and [13] describe a data mining framework for adaptively building intrusion detection models.[14] and [15] proposed a distributed intrusion detection framework based on autonomous mobile agents.[16] uses a layered framework to build a network IDS which can detect a wide variety of attacks reliably and efficiently when compared to the traditional network IDS but the accuracy of less occurring attack is not good.

Therefore, an attempt is made in this paper to build an IDS by integrating layered framework with neural network so as to combine the advantages of both the approaches. Thus, an integrated IDS is proposed which can detect a wide variety of attacks with less false alarm rate and can operate efficiently in high speed network.

III. PROPOSED INTEGRATED IDS ARCHITECTURE

In this paper, two architectures are proposed for integrated IDS which we call as Model A and Model B. Model A consider all features of training dataset and Model B consider only those features which contribute to classification process so as to reduce the computation time. Model A and Model B are explained below:

A. Proposed IDS based on layered framework integrating with neural network using all features (Model A)

IDS under consideration combine the advantages of both layered framework and neural network The proposed IDS is used to detect four common types of attacks like Denial of Service(DoS),Probe, Remote to Local(R2L),User to Root (U2R) and normal records also. Thus, IDS is divided into four layers which are used to classify attacks as mentioned in Fig.1.

Each layer of IDS consist of three components:

1) Data preprocessor

This component is used to collect the data from desired source. Here, KDD cup 99 dataset is used which is publicly available [18].

2) Encoder

Encoder is basically used to encode the data into desired format. The attribute given in KDD data set are converted into double data type to make it compatible with ANN Tool Box of Matlab.

3) Classifier

This component is used to analyze the audit pattern and classify it to detect attacks. Here, Layered framework integrated with back propagation neural network (BPN) with 'trainscg' as training algorithm is used to classify the records as normal or attack.

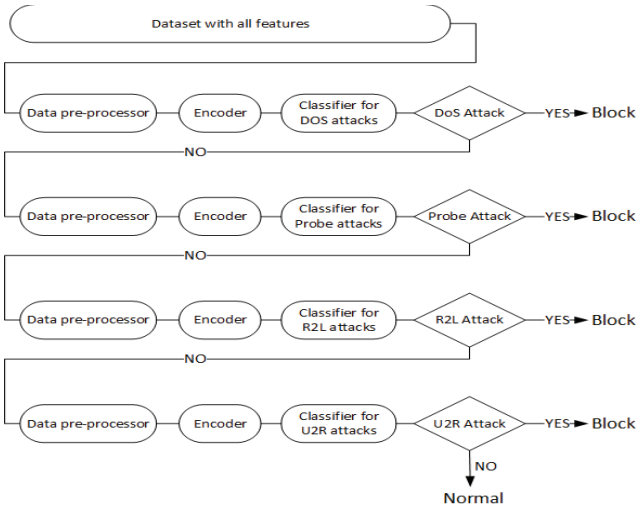

Fig.1. Architecture of proposed IDS based on layered framework integrated with neural network

B. Proposed IDS based on layered framework integrated with neural network using feature extraction(Model B)

IDS under consideration integrate both layered framework and neural network same as above IDS but now the dimensionality of dataset is reduced so that only those features are used that contribute to the classification process. This architecture of IDS is shown in Fig.2.

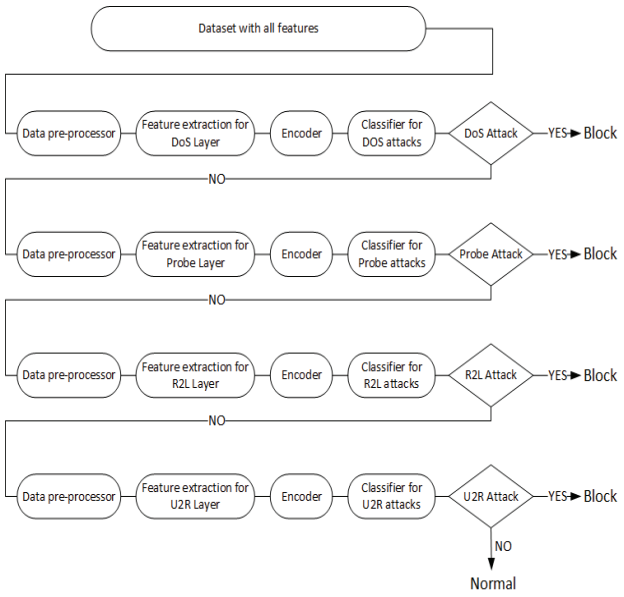

Fig.2. Architecture of proposed IDS integrating layered framework with neural network using feature extraction

In this architecture data preprocessor not only collect the data but also perform the task of data cleaning by extracting features for each layer using Principal Component Analysis (PCA) method as mentioned in [17]. Since the proposed IDS comprises of four layers corresponding to each attack so PCA is applied to individual layer and results are shown in Table I.

TABLE I. FEATURE EXTRACTION USING PCAFOR EACH LAYER

Layer no.	Type of attack	No of feature extracted using PCA
1	DoS	8 (f1,f2----f8)
2	Probe	11 (f1,f2----f11)
3	R2L	11 (f1,f2----f11)
4	U2R	10 (f1,f2----f10)

IV. DATA SET DESCRIPTION

Experiments are conducted using KDD Cup 99 dataset [18] which consist of a set of 41features (Appendix A) derived from each connection and a label which specifies the connection records aseither normal or specific attack type. This databasecontains a standard set of data to be audited, whichincludes a wide variety of intrusions simulated in amilitary network environment. Generally the attacks fall into four main categories namely DoS, Probe, R2L and U2R. kdd-cup.data_10_percent.gzis used as training and validation dataset havingexactly 494,021 instances with 22 attack types as shown in Table II and corrected.gz as test dataset having exactly 311,029 instances as shown in Table III.

TABLE II. TRAINING DATA SET

DoS (391458)	U2R (52)	Probe (4107)	R2L (1126)
Back	Buffer-overflow	Ipsweep	Ftp-write
Land	Load Module	Nmap	Guess-passwd
Neptune	Perl	Portsweep	Imap
Pod	Rootkit	Satan	Multihop
Smurf			Spy

TABLE III. TEST DATA SET

DoS (229853)	U2R (70)	Probe (4166)	R2L (16349)
Back	Buffer-overflow	Ipsweep	Ftp-write
Land	Load Module	Nmap	Guess-passwd
Neptune	Perl	Portsweep	Imap
Pod	Rootkit	Satan	Multihop
Smurf	Ps	Mscan	Spy
Teardrop	Sqlattack	Saint	Warezclient
Apache 2	Xterm		Warezmaster
Mailbomb			Phf
Process Table			Httptunnel
Udpstorm			Named
			Snmpgetattack
			Xlock

Test data is also labeled as either normal or as one of the attacks belonging to the four attack classes. It is important to note that the test data includes specific attacks which are not present in the training data. This makes the intrusion detection task more realistic.

V. EXPERIMENTAL RESULTS

Proposed IDS is simulated to obtain results using Matlab2008 version. All experiments are done on Intel® Core 2 Duo CPU P9600 @ 2.66 GHz having 8GB of RAM. The operating system used is Windows 7. Simulation is performed using data set described in Section IV.

A. Experimental setup for Model A

Experimental neural network setup for DoS Layer is shown in Figure 3. Similar network is setup for every layer of the IDS.

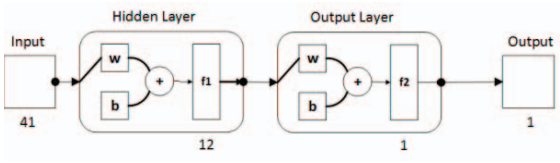

Fig. 3. Experimental neural network setup for DoS Layer (Model A)

The input layer consists of 41 neurons representing 41 attributes in the KDD data set.Hidden layer comprises of 12 neurons that give the optimal result. Since the layer is dedicated to detect DoS attack thus only 1 output neuron is used for DoS(1) and normal(0) records. After the process of data cleaning, preprocessing and encoding data is fed to above mentioned model.Each layer detects the individual attack dedicated for and pass the remaining record (non attack) to next layer for classification. Number of records remained after filtering from each layer is declared as normal.

During training process, training dataset is divided into training(70%),validation(15%) and testing (15%) data subsets. Training of the network is stopped when mean squared error(mse) on the validation set is constant for 6 epochs. Trained network is then tested on testing dataset for level 1 testing. Numberof iterations, time and training dataset used to train each layer is mentioned in Table IV.

TABLE IV. SIMULATION RESULTS OF MODEL A

Layer no.	Type of attack	Training dataset	Number of Iterations	Time in minutes
1	DoS	5000 (DoS) + 15278 (non DoS)	252	2.23
2	Probe	4107 (Probe) + 16177 (non Probe)	137	1.27
3	R2L	3500 (R2L) + 10070 (non R2L)	112	0.45
4	U2R	52 (U2R) + 1800 (non U2R)	44	0.03

The total time taken to train the particular network is 3.98 minutes approximately and it took 545 iterations to train the network. Figure 4shows the performance of the DoS layer taking into account training, validation and level 1 testing. The training process is repeated 252 times as training stops only when mean squared error (mse) on the validation set is constant for 6 epochs. It is clearly shown that mseis constant after epoch 246 thusthe best validation performance is 0.001629 at epoch no. 246.

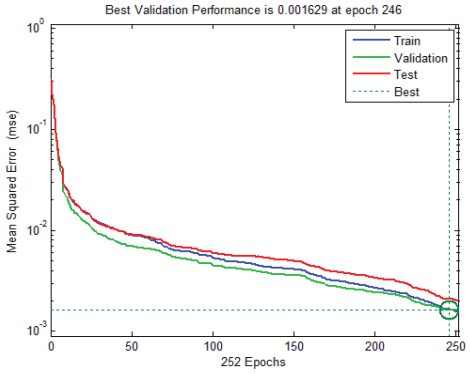

Fig. 4. Performance of DoS Layer (Model A)

Fig.5. shows the performance of the probe layer taking into account training, validation and level 1 testing.Similar to DoS layer the training process stops at epoch 137 asmean squared error on the validation set is constant for 6 epochs. It shows that the best validation performance is 0.0021412 at epoch no. 131.

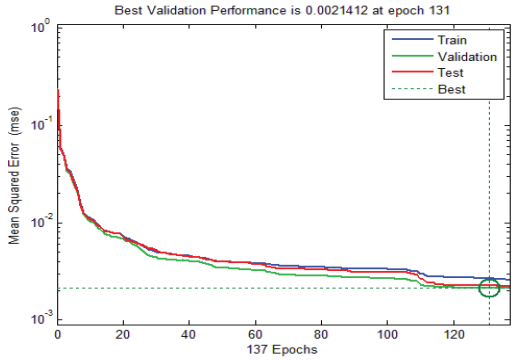

Fig.5. Performance of Probe Layer (Model A)

Fig.6. shows the performance of the R2L Layer taking into account training, validation and level 1 testing. It shows that the best validation performance is0.013334 at epoch no. 108 and training stops when mse on validation set get constant.

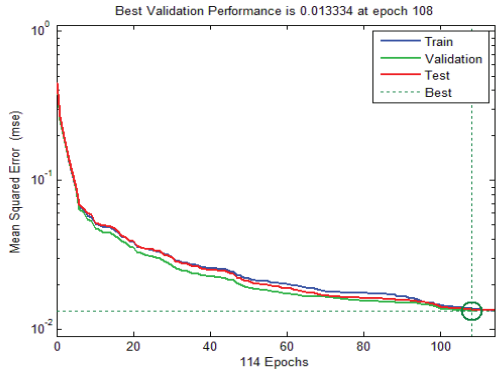

Fig. 6. Performance of R2L Layer (Model A)

Fig.7. shows the performance of the U2R Layer taking into account training, validation and level 1 testing. It shows that the best validation performance is 0.0090596 at epoch no. 38 and the training stops at epoch 42 when mse get constant.

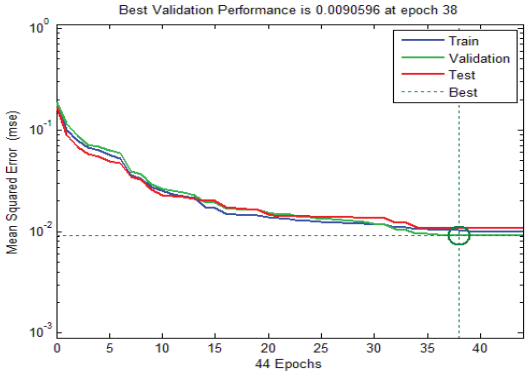

Fig.7. Performance of U2R Layer (Model A)

Model A is tested for level 2 testing using a new set of data. Testing dataset consist of 50000 records randomly chosen from KDD cup99 test dataset containing 311029 records. The new confusion matrix as shown in Figure 8 showed a success rate of 97% .Confusion matrix is a ranking method applied to any kind of classification problem. The size of matrix is determined by the number of distinct classes that are to be detected. Horizontal axis represent actual class, vertical axis represent predicted class and diagonal element represent all the correct classification.

Fig.8. Confusion Matrix for Model A

Fig.8. shows that Probe attacks identified by layer 2 (Probe layer) is 2608(86.96%) but DoS layer detect 235(7.8%) probe attack,R2L layer detect 20(0.006%) probe attack so total identified probe attacks are 2862(95.45%) instead of 2608. Similarly total no. of detected attack and normal is calculated using Confusion matrix of Figure 8 and tabulated in Table V.

TABLE V. ACCURACY OF INDIVIDUAL ATTACK (MODEL A)

	% Detection					(Total % Blocked)
	DoS	*Probe*	*R2L*	*U2R*	*Normal*	
DoS	99.49	0.0004	0	0	0.0006	99.54
Probe	7.8	86.96	0.006	0	0.014	95.46
R2L	0	0.0173	97.84	0	23.04	99.58
U2R	0.033	0	63.3	23.33	0	90.00
Normal	0.0006	0.014	0.23	0	76.48	24.52

B. Experimental neural network setup for Model B

Proposed IDS is divided into four layers which is used to classify attack as mentioned in Figure 2.Experimental neural network setup for DoS layer is shown in Fig.9.

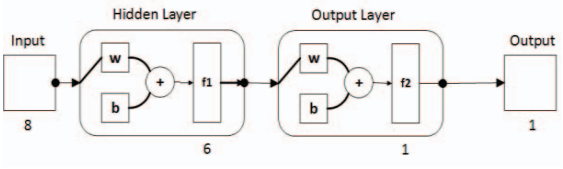

Fig.9. Experimental neural network setup for DoS layer(Model B)

The input layer consists of 8 neurons representing 8 features [A1] of training dataset of KDD data set. Hidden layer comprises of 6 neurons that gives the optimal result. Since the layer is dedicated to detect DoS attack thus only 1 output neuron is used for DoS and normal records. Similarly the experimental neural network set up is designed for other mentioned layers varying the no. of input neurons according to features extracted. After the process of data cleaning, preprocessing and encoding data is fed to classifier as shown in Figure 2.Each layer detect the individual attack dedicated for and pass the remaining record (non attack) to next layer for classification. Number of records remained after filtering from each layer is declared as normal. Number of iterations, time and training dataset used to train each layer is mentioned in Table VI.

Table VI. Simulation Result of Model B

Layer no.	Type of attack	Training dataset	No. of Iteration	Time in min.
1	DoS	5000 (DoS) + 15278 (non DoS)	111	1.00
2	Probe	4107 (Probe) + 16177 (non Probe)	54	0.27
3	R2L	3500 (R2L) + 10070 (non R2L)	184	1.05
4	U2R	52 (U2R) + 1800 (non U2R)	42	0.02

The total time taken to train the Model B is 2.34 min. approximately and it took 391 iterations to train the network. Fig.10. shows the performance of the DoS layer taking into account training, validation and level 1 testing. The training process is repeated 111 times as training stops only when mean squared error (mse) on the validation set is constant for 6 epochs. It is clearly shown that mse is constant after epoch 105 thus the best validation performance is 0.062695 at epoch no. 105.

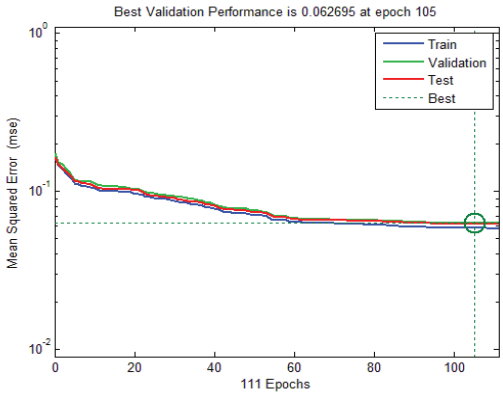

Fig.10. Performance of DoS Layer (Model B)

Fig.11. shows the performance of the probe layer taking into account training, validation and level 1 testing. Similar to DoS layer, it shows that the best validation performance is 0.070061 at epoch no. 48.

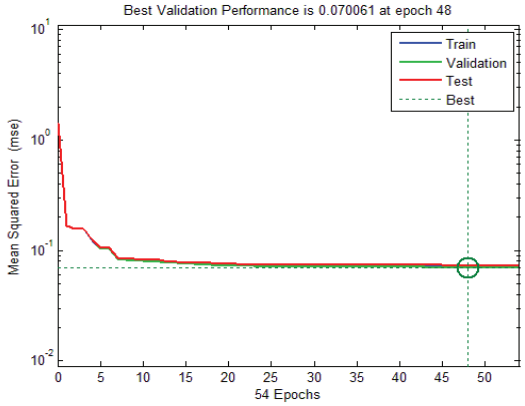

Fig.11. Performance of Probe Layer (Model B)

Fig.12. shows the performance of the R2L layer taking into account training, validation and level 1 testing. Similar to above layers, it shows that the best validation performance is 0.047996 at epoch no. 178.

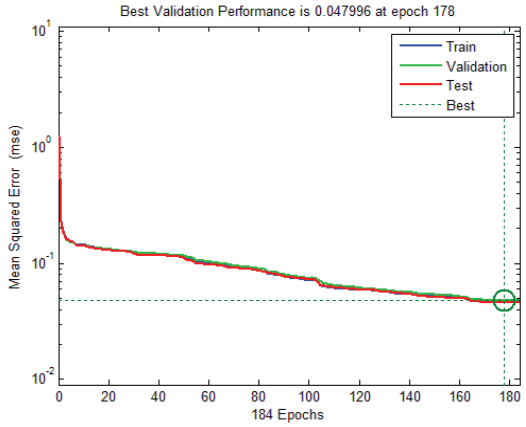

Fig.12. Performance of R2L layer (Model B)

Fig.13. shows the performance of the U2R layer taking into account training, validation and level 1 testing. It shows that the best validation performance is 0.014709 at epoch no. 36.

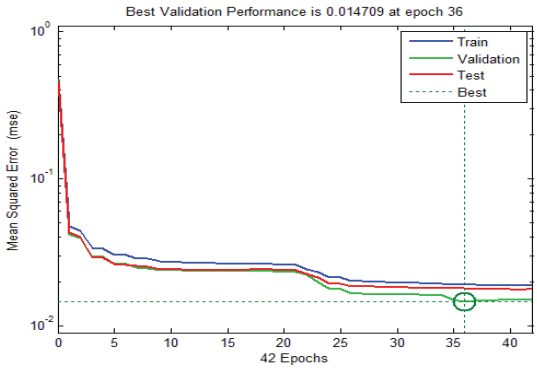

Fig.13. Performance of U2R Layer (Model B)

Model B is tested for level 2 testing for the same set of data consist of 50000 records as used for the first experimental setup. The confusion matrix as shown in Figure 14 showed a success rate of 94.4% and accuracy of individual attack is mentioned in Table VII.

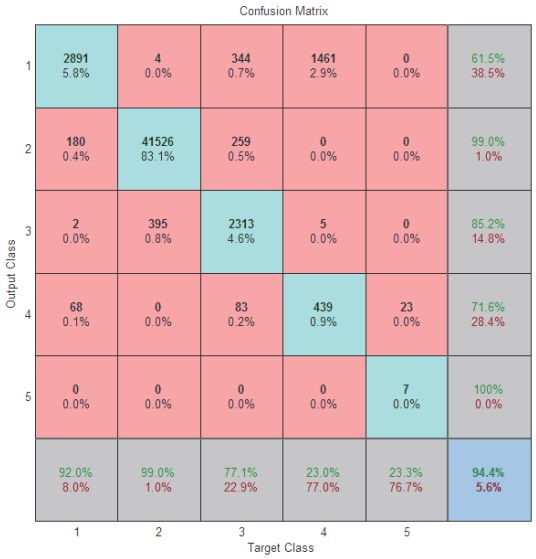

Fig.14. Confusion Matrix for Model B

Fig.14. shows that U2R attacks identified by layer 5 (U2R layer) is 7 (23.33%) but R2L layeralso detect 23 (76.66%) U2R attack so total U2R attack is 30(100%) instead of 23.33% .Similarly total blockage of each attack and normal is calculated using confusion matrix of Figure 14 and results are tabulated in Table VII.

TABLE VII ACCURACY OF INDIVIDUAL ATTACK OF MODEL B

	% Detection					(Total % Blocked)
	DoS	*Probe*	*R2L*	*U2R*	*Normal*	
DoS	99.04	0.0094	0	0	0.0001	99.99
Probe	0.0863	77.12	0.0276	0	11.4	77.2
R2L	0	0.0001	23.04	0	76.69	23.05
U2R	0	0	76.66	23.33	0	100
Normal	0.0573	0.0001	0.0216	0	92.04	0.0795

The results of Model A and Model B are compared with existing approaches and a summary is shown in Table VIII.

TABLE VIII. COMPARISON OF PROPOSED IDS WITH EXISTING APPROACHES

Approach	DoS	Probe	R2L	U2R	Success rate (%)
Neural Network(BPN only)	-	-	-	-	73.9
Layered framework with Conditional Random Field	97.40	98.62	29.62	86.3	90.7
Proposed Layered Framework integrated with Neural Network using all features (Model A)	99.54	95.46	99.58	90	97.1
Proposed Layered Framework integrated with Neural Network using	99.99	88.52	23.05	100	94.3

Feature Extraction (Model B)					

Result shows that Model A increases attack detection accuracy significantly and perform better than all existing approaches. Model B is proposed so as to increase the accuracy of attack detection in less time. The results of Model B shows although overall success rate decreases yet detection rate of DoS (most occurring attack) and U2R increases by 2.3% and 13.7% respectively when compare with layered framework integrated with CRF. The result further shows that there is significant reduction in time also when Model B is used as Model A train network in 3.98 minutes but Model B completes training in 2.34 minutes.

VI. CONCLUSION

In this paper, Intrusion detection system (Model A and Model B) are designed by integrating layered framework with neural network. It is observed that Model A which consider all features of training dataset attains high accuracy while Model B which consider feature extraction reduces training time but with a slight decrease in success rate of attack detection. Results of Model A and Model B suggest that proposed IDSs works effectively for detecting various attack in the network.

From practical point of view, the experimental results imply that there is still scope of improvement as the proposed systems are not able to detect all types of attacks, thus it is interesting to investigate in this direction.

REFERENCES

[1] SANS InstituteInfoSec Reading Room. http://www.sans.org/reading_room/whitepapers/detection/understanding-intrusion-detection-systems_337.

[2] Herv´e Debar, Monique Becke, Didier Siboni, "A Neural Network Component for an Intrusion detection system," Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 240–250,1992.

[3] Suseela T. Sarasamma, Qiuming A. Zhu, Julie Huff, "Hierarchical Kohonenen Net for Anomaly Detection in Network Security," IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics, vol. 35(2), 2005.

[4] J. Ryan, M. Lin, R. Mikkulainen, "Intrusion Detection with Neural Networks," Advances in Neural Information Processing Systems, vol. 10, 1998, MIT Press.

[5] R. Lippmann, R. Cunningham, "Improving Intrusion Detection Performance using Keyword Selection and Neural Networks," RAID Proceedings, West Lafayette, Indiana, Sept 1999.

[6] A. Ghosh, A. Schwartzbard, "A study in using Neural Networks for Anomaly and Misuse Detection, " Proceedings of the 8th USENIX Security Symposium, 1999.

[7] Shi-Jinn Horng, Pingzhi Fan, Yao-Ping Chou, Yen-Cheng Chang and Yi Pan, "Feasible Intrusion Detector for Recognizing IIS Attacks Based on Neural Networks," Computer & Security, vol.27, pp. 84-100, 2008.

[8] Iftikhar Ahmad, Azween B Abdullah,Abdullah S Alghamdi , "Application of Artificial Neural Network in Detection of Probing Attacks," IEEE Symposium on Industrial Electronics and Applications , pp. 557-562, 2009.

[9] I Mukhopadhyay, M Chakraborty, S Chakrabarti, T Chatterjee,"Back Propagation NeuralNetwork Approach to Intrusion Detection System," International Conference on Recent Trends in Information Systems, 2011.

[10] Saso Dzeroski, Bernard Zenko, "Is Combining Classifiers Better than Selecting the Best One," Machine Learning vol. 54(3), pp. 255-273, 2004.

[11] Chuanyi Ji, Sheng Ma, "Combinations of Weak Classifiers," IEEE Transactions on Neural Networks, vol.8(1) , pp. 32–42 ,1997.

[12] Paul Dokas, Levent Ertoz, Vipin Kumar, Aleksandar Lazarevic, Jaideep Srivastava , Pang-Ning Tan, "Data Mining for Network Intrusion Detection," Proceedings of the NSF Workshop on Next Generation Data Mining, pp. 21–30, 2002.

[13] Wenke Lee, Salvatore J. Stolfo , Kui W. Mok, "A Data Mining Framework for Building Intrusion Detection Model," Proceedings of the IEEE Symposium on Security and Privacy, pp. 120–132 , 1999.

[14] Dalila Boughaci, Habiba Drias, Ahmed Bendib, Youcef Bouznit , Belaid Benhamou, "Distributed Intrusion Detection Framework Based on Mobile Agents," Proceedings of IEEE International Conference on Dependability of Computer Systems, pp. 248–255, 2006.

[15] Jai Sundar Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene H. Spafford ,Diego Zamboni, "An Architecture for Intrusion Detection Using Autonomous Agents," Proceeding of IEEE 14th Annual Computer Security Applications Conference, pp. 13–24, 1998.

[16] Kapil Kumar Gupta, Baikunth Nath, Ramamohanarao Kotagiri, "Layered Approach Using Conditional Random Field for Intrusion Detection," IEEE Transactions on Dependable and Secure Computing, vol. 7(1), pp. 35-49, March,2010.

[17] Gopi K. Kuchimanchi, Vir V. Phoha, Kiran S. Balagani, Shekhar R. Gaddam,"Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems," Proceedings of the 2004 IEEE Workshop on Information Assurance and Security, June 2004.

[18] KDD Cup 1999 Intrusion Detection Data. http://kdd.ics.uci.edu/databases/ kddcup99/kddcup99.html(accessed on 20-01-2011).

APPENDIX A

TABLE A.1 KDD CUP 99 DATA SET

Feature Number	Feature Name	Description
1	duration	length of connection
2	protocol_type	type of protocol
3	service	network service on data set
4	Flag	normal or status of the connection
5	src_bytes	number of data bytes from source to destination
6	dest_bytes	number of data types from destination to source
7	Land	1 if connection is from the same host/port; 0 otherwise
8	wrong_fragment	number of wrong fragments
9	urgent	number of urgent packet
10	hot	number of "hot" indicator
11	num_failed_logins	number of failed login attempts
12	logged_in	1 if successfully logged in,0 otherwise
13	num_compromised	number of compromised conditions
14	root_shell	1 if root shell is obtained; 0therwise
15	su_attempted	1 if ``su root'' command attempted; 0 otherwise
16	num_root	number of ``root'' accesses
17	num_file_creations	number of file creation operations
18	num_shells	number of shell prompts

19	num_acess_files	number of opearation on acess control files
20	num_outbound_cmds	number of outbound commands in an ftp session
21	is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise
22	is_guest_login	1 if the login is a ``guest'' login; 0 otherwise
23	count	number of connections to the same host as the current connection in the past two seconds
24	srv_count	number of connections

Feature Number	**Feature Name**	**Description**
25	serror_rate	% of connections that have ``SYN'' errors
26	srv_serror_rate	% of connections that have ``SYN'' errors
27	rerror_rate	% of connections that have ``REJ'' errors
28	srv_rerror_rate	% of connections
		that have ``REJ'' errors
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	srv_diff_host_rate	% of connections to different hosts
32	dst_host_count	count for destination host
33	dst_host_srv_count	srv_count for destination host
34	dst_host_same_srv_rate	same_srv_rate for destination host
35	same_srv_rate	diff_srv_rate for destination host
36	dst_host_same_src_port_rate	same_src_port_rate for destination host
37	dst_host_srv_diff_host_rate	diff_host_rate for destination host
38	dst_host_serror_rate	serror_rate for destination host
39	dst_host_srv_serror_rate	srv_serror_rate for destination host
40	dst_host_rerror_rate	rerror_rate for destination host
41	dst_host_srv_rerror_rate	srv_serror_rate for destination host