Detecting DOS Attacks with Artificial Neural Networks

Tarek Ibrahim H. Melad

Abstract-- **The attack detection tools are very important for providing safety in computer and network system. These tools fully depend on accuracy of attack detection. Moreover, the detection is also must for prevention of any attack. Therefore accurate detection of attack is very important. A number of attempts have been done in the field of attack detection but they suffered many limitations such as time consuming statistical analysis, regular updating, non-adaptive, accuracy and flexibility.**

Therefore, it is an artificial neural network that supports an ideal specification of an attack detection system and is a solution to the problems of previous systems. As a result, an artificial neural network· inspired by nervous system has become an interesting tool in the applications of attack detection systems due to its promising features. Attack detection by artificial neural networks is an ongoing area and thus interest in this field, has increased among the researchers.

Keywords-- **Detect, attack, artificial neural networks, Intrusion Detection Systems**

I. Advantages And Drawbacks

ANNs have several desirable properties for intrusion detection, their ability to learn complex patterns in data sets and generalize from known patterns to new makes them successful for both misuse and anomaly detection. Other benefits include their flexibility with respect to noisy/missing data and that some networks are capable of continuously learning during run-time. Initially, statistical techniques were used to perform anomaly detection. However, ANNs now provide a good alternative. Drawback of a statistical component is that assumptions need to be made on the distribution of the data, which is not the case with ANNs. Furthermore, ANNs are less sensitive to the selected input data (features), i.e., if a feature is irrelevant, the ANN is capable of learning to ignore it.

However, aside from the fact that ANNs are black boxes, main drawbacks of using ANNs are:

- Determining the topology of the ANN is difficult and time consuming; mostly done ad hoc or optimized with an evolutionary algorithm.
- Large amounts of training data are necessary, and the performance of the network is directly dependent on this, which requires elimination of unnecessary features used for training.

II. Determining The ANN Architecture

There is no certain mathematical approach for obtaining the optimum number of hidden layers and their neurons. For choosing optimum set of hidden layers and its numbers of neuron a comparison is made for many cases and optimum is selected as shown in the table 10. In our experiment 4 layers MLP with two hidden layers is found to be optimum among several cases.

The topology of the ANN is important, not only to the performance of the network, but also for how the intrusion detection is achieved. For example, there are many different ways in which an ANN can output its classification, which gives different properties to the intrusion detection.

The design of our system consists of one input, two hidden and one output layer. The Input layer takes input from the input file that; contains data for training of the net. The hidden layers take inputs from the outputs of the input layer and apply its activation function. After this the output is sent to the output layer. The output layer allows a neural network to write output patterns in a file that are used for analysis of intrusion. The general architecture of our system is shown in figure 1.

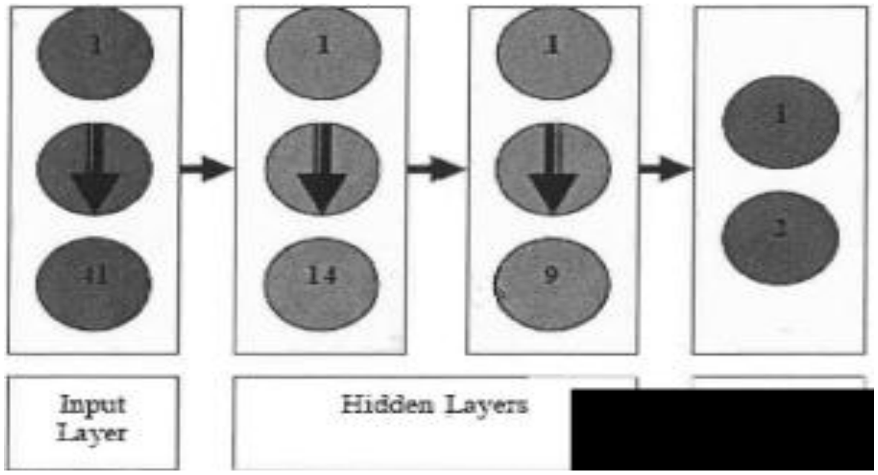

Figure1 : General Architecture of ANN

The comprehensive structural design of attack detection system for normal and attacked scenario is shown below in figure 2.

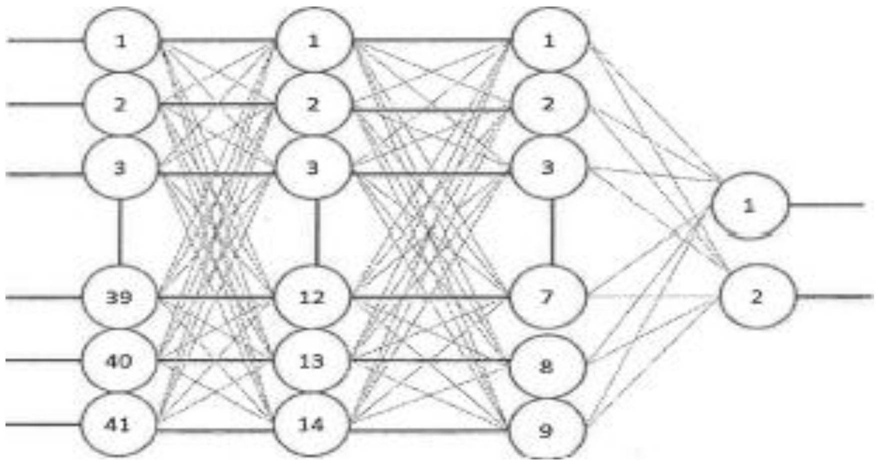

Figure 2: a comprehensive design ofIDS for normal scenario

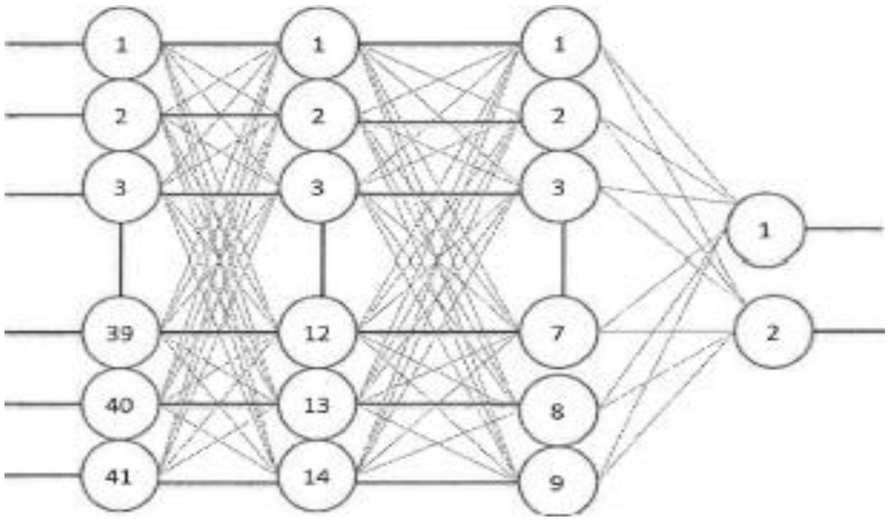

Figure 3: a comprehensive design ofIDS for attack scenario

III. Detecting DoS Attacks With Dedicated MLPs

Experiments needed to complete the dissertation using MATLAB Neural Network Toolbox which was used for the implementation of the MLP networks. Using these tools one can define specifications like number of layers, number of neurons in each layer, activation functions of neurons in different layers. Then the training feature vectors and the corresponding desired outputs can be fed to the neural network to begin training.

Experiments were conducted as described below:

- Creating testing and training vectors using Text Scan Tool;
- Determining the topology of ANN;
- Training the ANN with the original 41 feature 10% data set, and testing it for DoS category detection with the 41 feature testing set.
- Performing the empirical feature reduction method, in order to obtain reduced set for DoS category.
- Training and testing the ANN with the full feature set, necessary feature set and union of necessary and important feature sets; trainings have been conducted with both full and 10% training sets.
- Comparing the results of DoS attack detection of ANNs trained with reduced sets with ANNs trained with the full feature set.

IV. Testing And Training Vectors

Training and testing vectors have been obtained from KDD cup99 data sets using Text Scan Tool. This tool, provided as freeware addition to MATLAB, imports the data from plain-text files into MATLAB data files.

Data has been imported as follows:

- Creation of matrix Trx which retains the 10% training set without labels .
- Creation of vector Try that consists of labels corresponding to traffic in Trx matrix .
- Creation of matrix TrXF which retains the full training set without labels .
- Creation of vector Tryp that consists of labels corresponding to traffic in TrXF matrix
- Creation of matrix T that corresponds to the testing set .

For the feature reduction following additional matrices is needed: 41 matrices that contain 40 features (one removed in each) from the 10% training set (vector Try corresponds to these matrices) and 41 matrices that contain 40 features (one removed in each) from the testing set). Finally, training and testing matrices containing 11 and 19 attributes derived from full and 10% training set, and testing set, respectively, are created. These matrices correspond to vectors Try and Tryp.

V. Determining The Topology Of ANN

Many authors have used a three layer neural network for classification of connection records in normal and attack classes. Most of such studies end up with a two class classifier that succeeds in classification of normal and attack records in 89-91 % of the cases. In yet another study, the authors used three and four layer neural networks and reported results of about 99.25% correct classification for their two class (normal and attack) problem.

The universal approximation theorem states that an MLP (with one or more hidden layers) can approximate any function with arbitrary precision and of course the price is an increase in the number of neurons in the hidden layer. The question is if anything is gained by using more than one hidden layer. One answer is that using more than one layer may lead to more efficient approximation or to achieving the same accuracy with fewer neurons in the neural network. One of the objectives of the present study is to evaluate the possibility of achieving better results with this less complicated neural network structure, in order to propose, a system which can be implemented in order to detect DoS attacks in real situations, such as corporate networks. U sfug a less complicated neural network is more computationally efficient. Also it decreases the training time. There is no certain mathematical approach for obtaining the optimum number of hidden layers and their neurons. For choosing optimum set of hidden layers and its number of neurons, a comparison is made for many cases and optimum is selected (a scenario which provides smallest root means square error) as shown in the Table 1.

Table 1.
Iterations in determining optimal structure

Iteration	Number of neurons in hidden layers	Root means square error
1	20, 10, 5	0.195
2	14, 9, 3	0.149
3	14, 9, 4	0.015
4	14,9	0.002
5	14, 10	0.010

ANN architecture

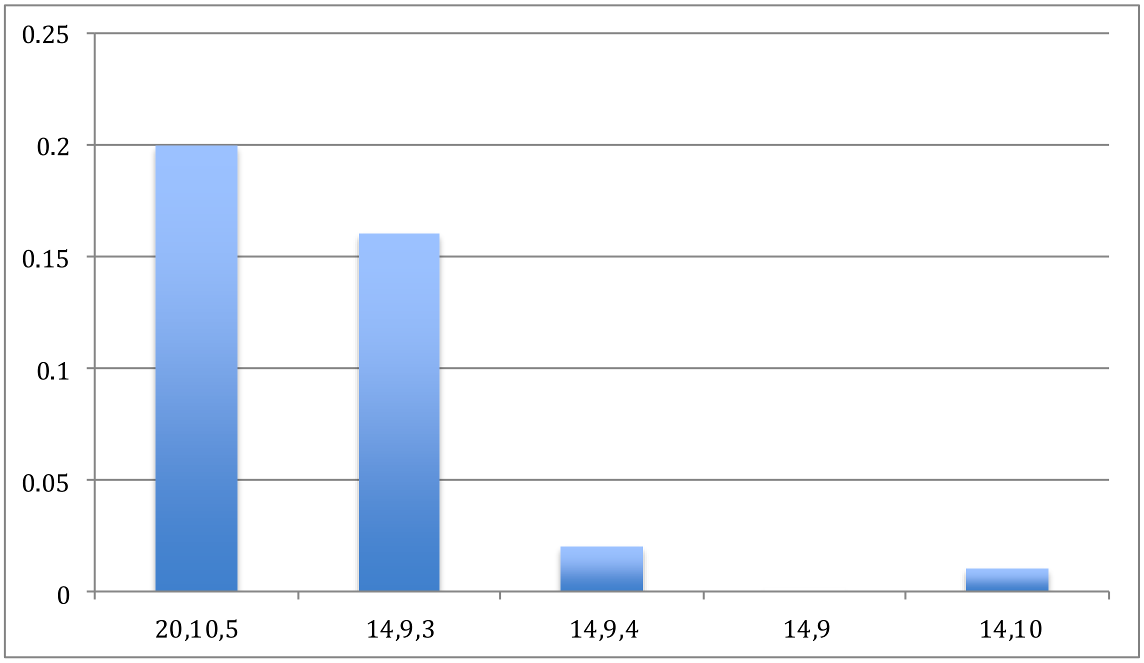

Figure 4 .Artificial Neural Network Structure

Therefore, experiments have been conducted on neural network with two hidden layers. Number of features used for training and testing dictate number of 41 input neurons. ANN in this experiment works as binary classifier, and so it has two output neurons, which will allow the further improvements to the model. Inner structure of neural network is set according to the table 1.

Choosing the training algorithm

Many studies have been conducted over choosing the proper ANN training algorithm. It has been noted that the resilient back-propagation shows best performance as compared to other ANN approaches towards attack detection. The system trained with this algorithm from both full feature set and reduced feature set shows high accuracy of DoS detection rate. Aside from that, resilient back propagation algorithm converges very quickly. It uses only the sign of the back-propagated gradient to change the biases/weights of the network. The basic principle is to eliminate the harmful influence of the size of partial derivative on the weight step.

The training phase consists of the following steps:

1. the feed-forward of input training pattern,
2. the calculation and back-propagation associated error and
3. the adjustment of the weights.

The aim of training means to get good responses to input that is similar but not identical.

Over-fitting problem

One problem that can occur during neural network training is over-fitting. In an over-fitted ANN, the error (number of incorrectly classified patterns) on the training set is driven to a very small value. However, when new data is presented, the error is large. In these cases, the ANN has memorized the training examples; however, it cannot generalize the solution to new situations.

One possible solution for the over-fitting problem is to find the suitable number of training epochs by trial and error. In this study, the training time is too long. Therefore, it is not reasonable to find the optimal number of epochs by trial and error.

A more reasonable method for improving generalization is called early stopping. In this technique, the available data is divided into three subsets. The first subset is the training set, which is used for training and updating the ANN parameters. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training similar to the training set error.

However, when the ANN begins to over-fit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights that produced the minimum error on the validation set are retrieved. In the present study, this training validation strategy was used in order to maximize the generalization capability of the ANN.

Empirical-based feature reduction

The feature ranking and selection problem for intrusion detection is similar in nature to various engineering problems that are characterized by:

Having a large number of input variables X = (X1, X2, ... , Xn) of varying degrees of importance to the output y; i.e., some elements of x are essential, some are less important, some of them may not be mutually independent, and some may be useless or irrelevant (in determining the value of y)

- Lacking an analytical model that provides the basis for a mathematical formula that precisely describes the input-output relationship, y = F (x)
- Having available a finite set of experimental data, based on which a model (e.g. neural networks) can be built for simulation and prediction purposes

Due to the lack of an analytical model, one can only seek to determine the relative importance of the input variables through empirical methods. A complete analysis would require examination of all possibilities, e.g., taking two variables at a time to analyze their dependence or correlation, then taking three at a time, etc. This, however, is both infeasible (requiring numerous experiments) and not infallible (since the available data may be of poor quality in sampling the whole input space).

Therefore, the technique of deleting one feature at a time to rank the input features and identify the most important ones for intrusion detection is applied in our dissertation. One input feature is deleted from the data at a time; the resultant data set is then used for the training and testing of the classifier. Then the classifier's performance is compared to that of the original classifier (based on all features) in terms of relevant performance criteria. The importance of the feature is ranked according to a set of predefined rules based on the classifier comparison.

To perform feature ranking, the classifier is trained with the 41-feature set. After that for each feature the following procedures are done:

1. Delete the feature from the (training and testing) data.
2. Use the 40-feature data set to train the classifier.

3. Analyze the performance of the classifier using the test set.
4. Rank the importance of the feature according to the rules.

Performance Metrics

To rank the importance of the 41 feature, three main parameters were considered: accuracy, training time and testing time. Each feature is ranked according to the predefined rules below that are applied to the result of comparison of the original 41-feature classifier and the 40-feature classifier. For example, if one attribute is necessary (which means it is very important for detection) then removing it from feature vectors would result in decreasing accuracy and increasing at least one time (be it training or testing) . Each feature will be ranked as "necessary", "Important but not necessary" and "not needed for Dos attacks" according to the following predefined rules that are applied to the result of performance comparison of the original 41-feature.

1. If a feature is necessary (which means it is very important for detection) then removing it from feature vectors would result in decreasing accuracy and increasing at least one time (training or testing time).
2. If a feature is not needed for detection, then removing it from the feature vector would result increasing accuracy or unchanged and decreasing at least one time (training or testing time).
3. In other situations, a feature is ranked as important but not necessary for detection of DoS attacks.

Reduced sets for DoS attacks and normal traffic categories

According to the methodology and rules of empirical-based reduction, after 41 experiments performed, a set of necessary, important and not needed attributes for DoS category detection have been extracted. Relevance of these attributes is shown in Table 2.

Table2:
Attributes ranks

#	Attribute name	Rank
1	Duration	Necessary
2	Protocol Type	Important, but not necessary
3	Service	Necessary
4	Flag	Not needed for DoS detection
5	Source bytes	Necessary
6	Destination bytes	Necessary
7	Land	Important, but not necessary
8	Wrong fragments	Necessary
9	Urgent	Important, but not necessary
10	Hot	Important, but not necessary
11	Logged in	Not needed for DoS detection
12	#compromised	Not needed for DoS detection
13	Root shell	Important, but not necessary
14	Su attempted	Not needed for DoS detection

15	#root	Not needed for DoS detection
16	#file creations	Important, but not necessary
17	#shells	Not needed for DoS detection
18	#access files	Necessary
19	#outbound cmds	Important, but not necessary
20	Is hot login	Not needed for DoS detection
21	Is guest login	Important, but not necessary
22	Count	Necessary
23	Srv count	Necessary
24	Serror rate	Necessary
25	Srv serror rate	Necessary
26	Rerror rate	Necessary
27	Srv rerror rate	Necessary
28	Same srv rate	Important, but not necessary
29	Diff srv rate	Important, but not necessary
30	Srv diff host rate	Not needed for DoS detection
31	Dst host count	Necessary
32	Dst host srv count	Necessary
33	Dst host same srv rate	Important, but not necessary
34	Dst host duff srv rate	Necessary
35	Dst host same src port rate	Necessary
36	Dst host srv diff host rate	Important, but not necessary
37	Dst host serror rate	Necessary
38	Dst host srv serror rate	Necessary
39	Dst host serror rate	Necessary
40	Dst host srv serror rate	Necessary

Comparing with the results gained by other feature reduction methods

Similar research has been conducted in several other researches, using support-vectors, LGPs, MARS and other methods. To validate the results, a subset of necessary attributes will be compared to results achieved by other authors.

Support-vector based methods usually extract smaller subsets. For example, with the SVDF function, as described in (Vemuri et al. 2006) most important features are:

- Count (number of connections made to the same host system in a given interval of time).
- Srv _ count (number of connections to the same service as the current connection during a specified time window).
- Dst_host_srv_serror_rate (percentage of connections to the same service that have SYN errors from a destination host).
- Errors _rate (percentage of connections that have SYN errors.
- dst_host_same_src_port_rate (percentage of connections to same service ports from a destination host).

The most important features, as extracted by LGPs (Linear Genetic Programming, Vemuri et al. 2006) are:

- count (number of connections made to the same host system in a given interval of time)
- num _ compromised (number of compromised conditions)
- wrong fragments (number of wrong fragments)
- land: binary decision (1 if connection is from/to the same host/port; 0 otherwise)
- logged in: binary decision (1 successfully logged in, 0 failed log-in)

The most important features, as extracted by MARS (Multivariate Adaptive Regression Splines, Vemuri et al. 2006) are:

- count (number of connections made to the same host system in a given interval of time)
- srv _ count (number of connections to the same service as the current connection during a specified time window)
- dst_ host_ srv _ diff_ host_rate (percentage of connections to the same service from different hosts to a destination host)
- source bytes (number of bytes sent from the host system to the destination system)
- destination bytes (number of bytes received by the source host from the destination host)

Comparatively speaking, empirical-based methodology of reduction adds more attributes as necessary to the reduced set. A brief overview of other reduced sets would lead to a conclusion that attribute 3, which describes a service, is necessary, as DoS attacks are based on target service, among others.

VI. TESTING MLP CLASSIFIERS

After the training is completed, the weights of the neural networks are frozen and performance of the neural networks evaluated. Testing the neural networks involves two steps, which are verification step and generalization step.

In verification step, neural networks are tested against the data which are used in training. Aim of the verification step is to test how well trained neural networks learned the training patterns in the training data set. If a neural network was trained successfully, outputs produced by the neural network would be similar to the actual outputs.

In generalization step, testing is conducted with data which not used in training. Aim of the generalization step is to measure generalization ability of the trained network. After training, the net only involves computation of the feed forward phase.

Testing proposed MLPs with empirical-based reduced sets

In this set, MLPs trained with the full feature set, necessary feature set and union of necessary and important (but not necessary) feature sets are tested. Testing has been performed on classifiers trained both with the complete training and 10% datasets.

Results, regarding overall accuracy, training and testing time compared to 41 feature full training set are shown in the Table 3. Times needed for training and testing with reduced feature sets are given compared to times needed to perform training and testing with the full featured sets. In the scenario 1, times needed to train and test are given as relative values to times needed to train the classifier with full test set. In the scenario 2, times are given relative to time needed to train classifier with 10% training set and test it with a testing set.

Table 3:
testing the proposed MLPs trained with reduced feature sets

Set	Accuracy	Training time	Testing time
Scenario 1 : full training set			
Full 41 feature set	95.2%	1	1
Necessary + important	97.2$%	0.71	1.04
Necessary feature set	98.5%	0.74	0.72
Scenario 2 : 10 % training set			
Full 41 feature set	93.1%	1	1
Necessary+ important	95.0%	0.79	1.12
Necessary feature set	96.7$%	0.80	0.79

Full training set

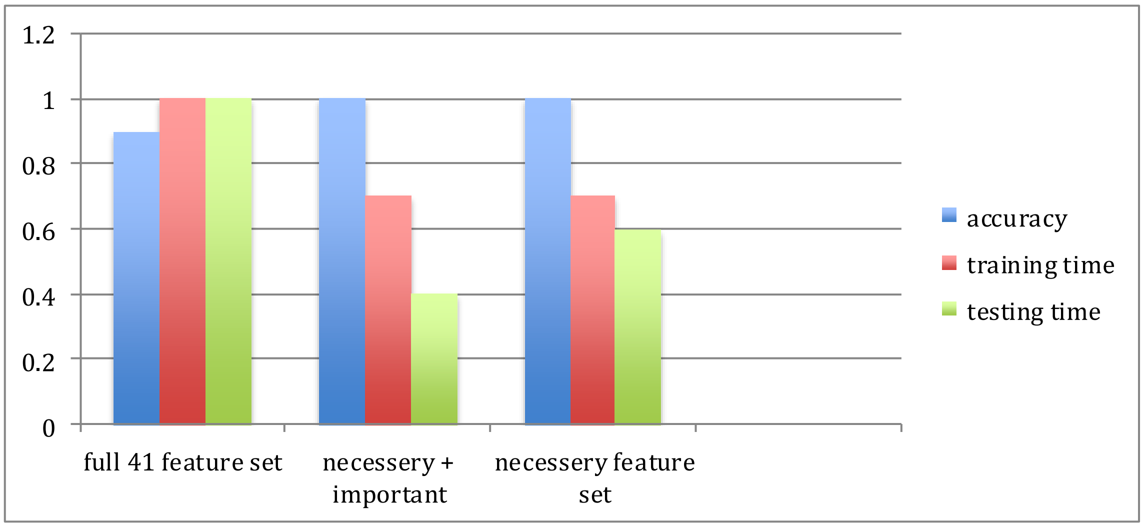

Figure 5 measuring the 3 matrices with full of dataset set

10% Training set

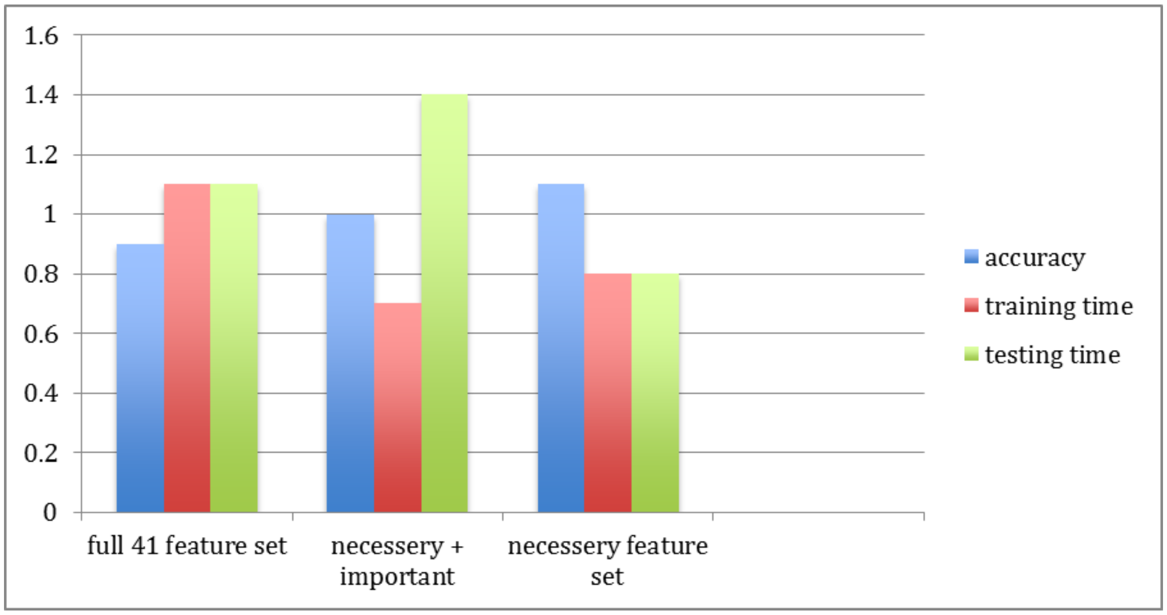

Figure 6 measuring the 3 matrices with 10% of datases

From the Table 3 the classifier that provides best results is classifier trained with the full training set using only necessary features. This classifier has best overall accuracy, provides the decision quickest and gets trained as second-best.

Next step in performance testing the artificial neural networks is examining detection accuracy and the number of false positives and false negatives that they generated for specific DoS attacks that are unknown for the classifier.

Hence, classifier is tested with the novel DoS attacks available only in the testing set: apache2, mail-bomb, processtable and udpstorm. Results of testing these attacks are given in Table 4 and figure7. Classifier used is the best classifier - trained with a full training set with only the necessary features.

Table 4:
Accuracy, false positive rate and false negative rate for specific DoS attacks

Attack	Accuracy	FP	FN
Apache2	98%	1%	1%
Mail-bomb	99%	0%	1%
Processtable	100%	0%	0%
Udpstorm	97%	1%	2%

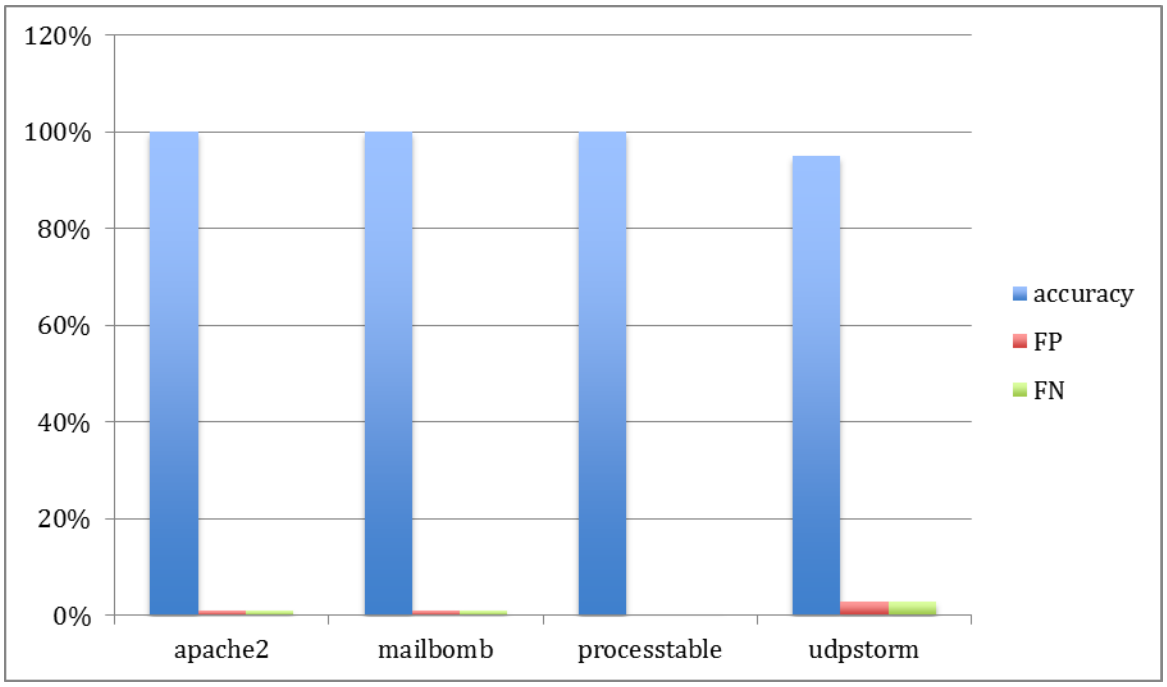

Figure 7 Accuracy,false positive rate and false negative rate for specific DoS attacks

The best classifier provides almost 100% accuracy with no false positives or negatives for the process table attacks, while making 1 % false positives and false negatives for apache2, 1 % false negatives for mail-bomb and 1 % false positives and 2% false negatives for udpstorm attacks.

On previously known attacks, this classifier provides almost 100% accuracy. This error in classification is inevitable consequence of feature reduction, yet due to speed gained and classification abilities of novel attacks, it can be concluded that classifier trained with proposed set is a very good neural network solution for detection of DoS attacks.

Comparing our proposed reduced set with reduced sets proposed by other authors

To prove the worthiness of our proposed DoS detection model, its results are compared to other training sets available before shown in table 5. There are 4 reduced sets of features with which classifiers are trained and tested: empirical-based reduced set with only necessary features from this research (having 20 necessary features) .,and other researches: SVDF, LGP and MARS based reduced sets (having 5 features).

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 8, August 2017)

Table 5:
Comparison of MLPs trained with different reduced feature sets found

Set	Accuracy	Training time	Testing time
SCENARIO 1 : full training set			
Empirical based necessary feature set	98.5%	0.74	0.72
SVDF reduced feature set	92.1%	0.66	0.61
LGP reduced feature set	93.2%	0.67	0.59
MARS reduced feature set	91.6%	0.73	0.71
SCENARIO 2 : 10% training set			
Empirical based necessary feature set	96.7%	0.80	0.79
SVDF reduced feature set	88.7%	0.72	0.68
LGP reduced feature set	89.4%	0.71	0.63
MARS reduced feature set	85.1%	0.75	0.70

FULL TRAINING SET

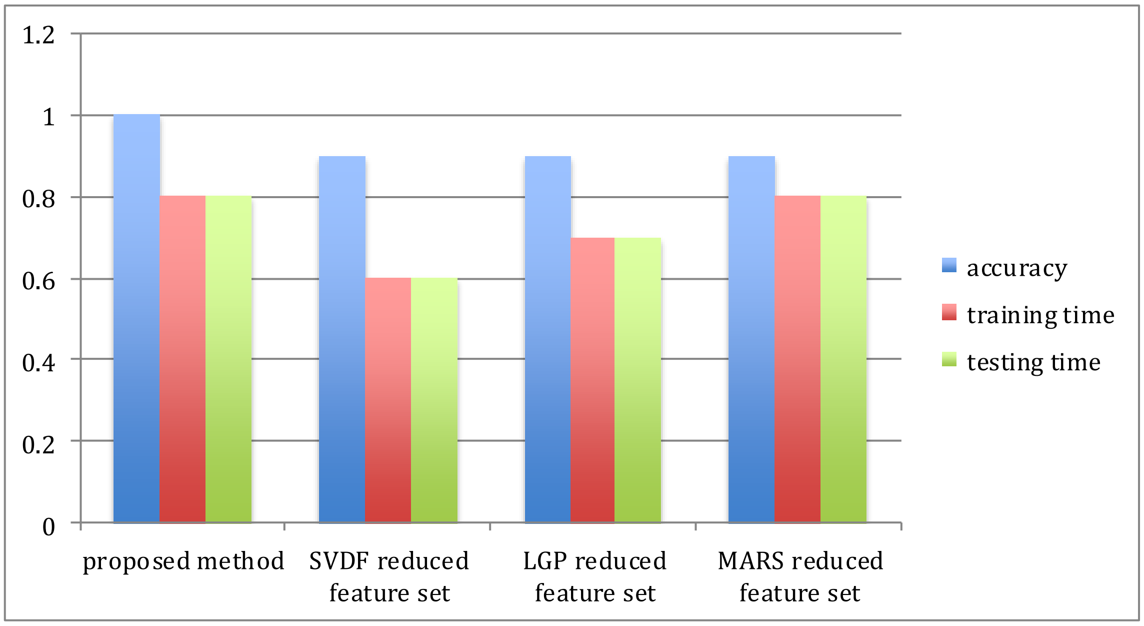

Figure 8 comparing our proposal with other proposals using full training of datasets

10% TRAINING SET

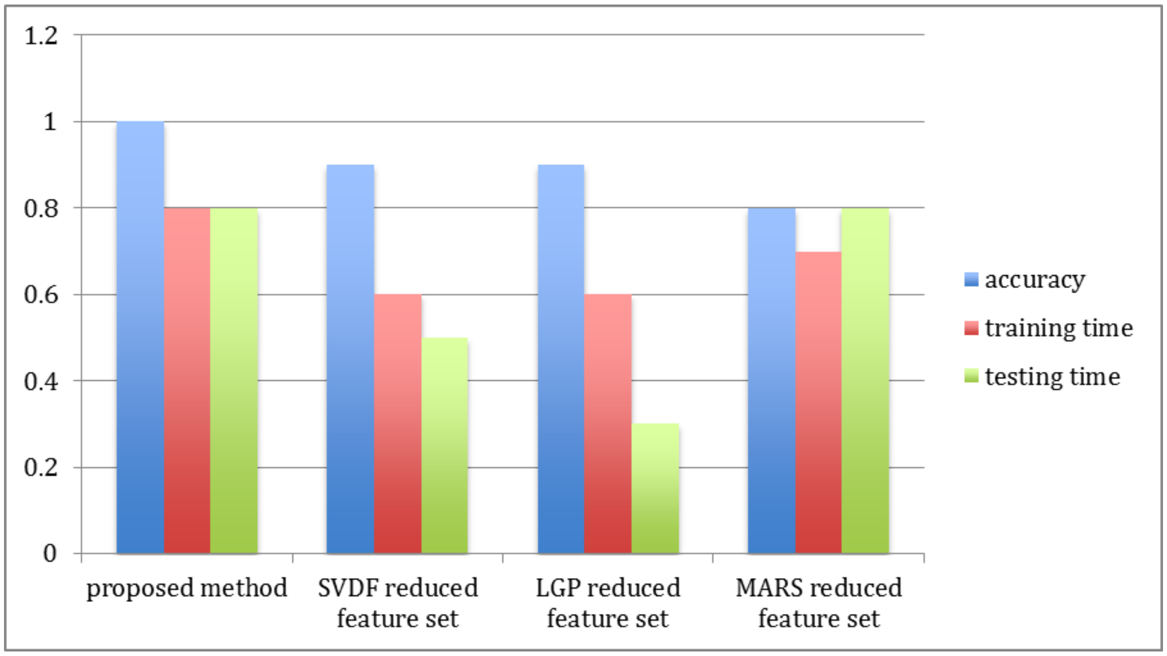

Figure 9 comparing our proposal with other proposals using 10% of the datas

VII. Summary Of Experimental Results.

A. We first described a general (i.e., independent of the modeling tools being used), performance based input ranking methodology: One input feature is deleted from the KDD cup99 dataset at a time; the resultant dataset is then used for the training and testing the classifier. Then the classifier's performance is compared to that of the original classifier (based on all features) in terms of relevant performance criteria. Finally, the importance of the feature is ranked according to a predefined set of rules based on the performance companion.
The procedure is summarized as follows:

- Compose the training set and the testing set;
- For each feature we did the following
- Delete the feature from the (training and testing) dataset;
- use the resultant data set to train the classifier;
- Analyze the performance of the classifier using the testing set, in terms of the selected performance criteria;
- Rank the importance of the feature according to the rules;

B. After choosing the necessary, important but not necessary and not needed features, according to the predefined rules. The proposed MLPs trained with full feature set necessary features and union of necessary and important (but not necessary) feature sets is tested.

C. Once the necessary, important but not necessary and not needed features, were ranked, the ANN classifier is trained and tested with the necessary features and union of necessary and important (but not necessary). Testing has been performed on classifiers trained both with the complete training and 10% datasets.

D. The results, regarding overall accuracy, training and testing time compared to 41 feature full training set . Times needed for training and testing with reduced feature sets are given compared to times needed to perform training and testing with the full featured sets.

E. The classifier that provides the best results is the classifier trained with the full training set using only necessary features. This classifier has the best overall accuracy, provides the decision I quickest and gets trained as second-best.

F. Examining the classifier and the number of false positives and false negatives that they generated for specific DoS attacks that are unknown for the classifier.

Hence, classifier is tested with the novel DoS attacks available only in the testing dataset i.e (apache2, mail-bomb, processtable and udpstorm), we found that the Classifier used is the best classifier trained with a full training set with only the necessary features as shown in table

G. Final optimized MLP IDS provides superior accuracy of 98.5%, substantially better than other referential IDS systems published up to now.

VIII. FUTURE WORK

The resourceful attack detection approach may be developed that have very low error rate and quick attack detection by using this approach with other neural networks in the form of hybrid architecture. And this approach can be used for detection other attacks.

IX. CONCLUSION

Security is needed through out distributed systems (interconnected components forming a network(min order to build dependable and trusted computing platforms. During the design phase of a distributed system, security policies are developed which account for the measures taken to ensure both the confidentiality and integrity of the system, when it is necessary. Confidentiality in this context refers to access constraints on users and is there to protect the data. The integrity refers to the correct running of the system and the data contained on the system.

Intrusion Detection Systems (IDSs) have become an important security tool for managing risk and an indispensable part of overall security architecture. An Intrusion Detection Systems gathers and analyzes information from various sources within computers and networks to identify suspicious activities that attempt to illegally access, manipulate and disable computer systems. Examples of IDSs are general network Intrusion Detection Systems, Web application firewalls, botnet-malware detection systems and many others.

REFERENCES

[1] Nagesh H.R, K. Chandra Sekran , Design and Development of proactive solutions for mitigating Denial of Service attacks, Proceedings of the 14th International conference on Advanced Computing and Communications, IEEE Press, India, Dec 2006, pp 157-162.

[2] Wei Chen , Dit-Yan Yeung , "Defending Against TCP SYN Flooding Attacks", International Conference on Systems and on Mobile Communications , Osaka, Japan, Nov. 2006.

[3] Ferguson. P, and D. Senie. "Network ingress filtering : Defeating denial of service attacks which employ IP source address spoofing." International Journal of Network management, Volume 15, Issue 1, 2005

[4] H. Yang, F. Ricciato, S. Lu, and L. Zhang, "Securing a wireless world," Proc. IEEE (Special Issue on Cryptography and Security Issues), vol. 94, no. 2, February 2006.

[5] M. Saleh and I. Al Khatib, "Throughput analysis of WEP security in ad hoc sensor networks," The Second International Conference on Innovations in Information Technology, Dubai, 2005.

[6] K. Agarwal and W. Wang, "Measuring performance impact of security protocols in Wireless Local Area Networks," The Second International Conference on Broadband Networks, Boston, USA, October 2005.

[7] M. Boulmalf, E. Barka, and A. Lakas, "Analysis of the effect of security on data and voice traffic in WLAN," Computer Communications, 30 (2007) 2468-2477.

[8] N. Baghaei, "IEEE 802.11 wireless LAN security performance using multiple clients," University of Canterbury, Christchurch, NZ.

[9] J. Wong, "Performance investigation of secure 802.11 wireless LANs:Raising the security bar to which level ?"University of Canterbury, Christchurch, NZ.

[10] H. Karl , A.Willig; 2005; John Wiley & Sons, Ltd.; PROTOCOLSAND ARCHITECTURESFOR WIRELESS SENSORNETWORKS; ISBN: 0-470-09510-5

[11] K.Sohraby, D.Minoli, T.Znati; 2007; John Wiley & Sons, Ltd.; Wireless sensor networks: technology, protocols, and applications; ISBN 978-0-471-74300-2

[12] J.Zheng, A.Jamalipour; 2009;John Wiley & Sons, Ltd.; WIRELESS SENSORNETWORKSA NetworkingPerspective;ISBN: 978-0-470-16763-2

[13] W.Dargie, C.Poellabauer; 2010; John Wiley & Sons, Ltd.; Fundamentals of wireless sensor networks : theory and practice;ISBN 978-0-470-99765-9

[14] B. Krishnamachari; 2005; CAMBRIDGE UNIVERSITY PRESS; Networking Wireless Sensors; ISBN-13 978-0-511-14055-6

[15] D.Sharma, S.Verma, K.Sharma; June 2013; Network Topologies in Wireless Sensor Networks: A Review; Available URL www.iject.org/vol4/spl3/c0116.pdf

[16] Barroso, U. Roedig, C. J. Sreenan, ‘‘Maintenance Awareness in Wireless Sensor Networks,’’ Short Paper, Proceedings. of the 1st European Workshop on Wireless Sensor Networks (EWSN), 2004.

[17] Deshpande et al., ‘‘Model-Driven Data Acquisition in Sensor Networks,’’ Proceedings of the 30th International Conference on Very Large Data Bases, 2004

[18] Krishnamachari, D. Estrin, S. Wicker, ‘‘Modelling Data-Centric Routing in Wireless Sensor Networks,’’ Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom'02), New York, June 2002.

[19] Wang, B. Li, K. Sohraby, ‘‘A Simple Mechanism on MAC Layer to Improve the Performance of IEEE 802.11 DCF,’’ Proceedings of the 1st Annual International Conference on Broadband, (Broadnets'04), San Jose, CA, Oct. 2004.

[20] Y. Chong et al., ‘‘Distributed Multitarget Multisensor Tracking,’’ in Multitarget Multisensor Tracking: Advanced Applications, Artech House, Norwood, MA, 1990.

[21] C.-Y. Chong, S. P. Kumar, ‘‘Sensor Networks: Evolution, Opportunities, and Challenges,’’ Proceedings of the IEEE, Vol. 91, No. 8, Aug. 2003

[22] Chiasserini, C.; Nordio, A.; Viterbo, E. On Data Scquisition and Field Reconstruction in Wireless Sensor Networks. In Proceedings of Tyrrhenian Workshop on Digital Communications, Sorrento, Italy, 2005.

[23] Culler, D.; Estrin, D.; Srivastava, M. Overview of sensor networks. IEEE Comput. Mag. 2004

[24] Dardari, D.; Conti, A.; Buratti, C.; Verdone, R. Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks. IEEE Trans. Mobile Comput. 2007

[25] Hao, J.; Brady, J.; Guenther, B.; Burchett, J.; Shankar, M.; Feller, S. Human tracking with wireless distributed pyroelectric sensors. IEEE Sensors J. 2006

[26] L. B. Ruiz, F. A. Silva, T. R. M. Braga, J. M. S. Nogueira, A. A. F. Loureiro, ‘‘On Impact of Management in Wireless Sensor Networks,’’ Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS’04), 2004

[27] Lee, D.-S.; Lee, Y.-D.; Chung,W.-Y.; Myllyla, R. Vital sign monitoring system with life emergency event detection using wireless sensor network. In Proceedings of IEEE Conference on Sensors, Daegu, Korea, 2006.

[28] M. Perillo and W. Heinzelman. DAPR: A protocol for wireless sensor networks utilizing an application-based routing cost. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2004.

[29] P. Bergamo, G. Mazzini, ‘‘Localization in Sensor Networks with Fading and Mobility,’’ Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC’02), 2002.

[30] Pering, T., T. Burd, and R. Brodersen, The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms. 1998.

[31] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys), 2003.

[32] Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys), 2003

[33] Verdone, R. Wireless Sensor Networks. In Proceedings of the 5th European Conference, Bologna, Italy, 2008.

[34] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 1999.

[35] Bivens, R. Gupta, I. Mclean, B. Szymanski, J. White, ‘‘Scalability and Performance of an Agent-Based Network Management Middleware,’’ International Journal of Network Management, 2004

[36] Bahl, P. and V. Padmanabhan, RADAR: An in-building RF-based user location and tracking system. IEEE Infocom, 2000.

[37] Tian, N. D. Georganas, ‘‘Connectivity Maintenance and Coverage Preservation in Wireless Sensor Networks,’’ Canadian Conference on Electrical and Computer Engineering, 2004

Appendix A

Preprocessed the dataset code load kddcup99.mat

```
attacks=unique(y)
%1  'back.'
%2 'buffer overflow.'
%3 'ftp_write.'
%4 'guess_passwd.'
%5 'imap.'
%6 'ipsweep.'
%7 'land.'
%8 'loadmodule.'
%9 'multihop.'
%10 'neptune.'
%11 'nmap.'
%12 'normal. '
%13 'perl.'
%14 'phf.'
%15 'pod.'
%16 'portsweep.'
 %17 'rootkit.'
%18 'satan. '
%19 'smurf.'
%20 'spy.'
%21 'teardrop.'
%22 'warezclient.'
 %23 'warezmaster.
for i=l:length(napadi)
s= strmatch (napadi { i}, y, 'exact') ;
t(i)=length(s); end
bar(t)
%choice of classes
 J1=11;
j2=15;
c1= attacks (jl);
c2= attacks (j2);
ic1= strmatch (cl, y, 'exact');
ic2= strmatch ( c2, y, 'exact');
```

```
ic12=union(icl,ic2);
% building of the training set
xclc2=x(ic12, :) ;
yclc2=y(ic12, :);
% Transformation of nominal features to numeric
 ss= [ ] ;
for i =1:3
ss.data{i}=unique(xsimb(:,i))
```